Kryptovaluutat ja ympäristö **Pelihistorian** podcastit **Ohjelmointiputka** 20 v.

SKROLLI 2022.1

SKROLLI

2022.1
SKROLLI.FI
9,90€

LIITTEENÄ PELILEHTI RETRO REWIND

TIETOKONEKULTTUURIN ERIKOISLEHTI

10. VUOSIKERTA

TIETOKONELEHTI HARRASTAJILLE

Prosessori: Avoin haastaja RISC-V
Ohjelmointi: Ada ja ATK-ajanlaskut
Historia: Tekstiviestianimaatiot

PAPERILTA DIGITAALISEEN

SÄHKÖISET MUISTIINPANOT
Craft
Day One
Drafts
Evernote
IA Writer
Milanote
Moleskine Smart
Note Air2
Obsidian
ReMarkable 2
Repaper
TransNote

Seuraavan sukupolven grafiikkarajapinta

VULKAN KÄYTÄNNÖSSÄ

Kulttuurikritiikissä

DISCO ELYSIUM

RETRO REWIND

Parempaa retrokuvaa

Alan Wake

Atari 2600:n Talvisota

Tietotekniikan kalevala

Janne Sirén
päätoimittaja

Elias Lönnrot kiersi Suomea ja Karjalaa vuosina 1828–1845 runonkeruumatkoillaan. Matkojen tavoitteena oli paikallisen kansanrunouden muistiinmerkitseminen ja julkaiseminen. Runoista syntynyt kansalliseepoksemme *Kalevala* taltioi ja nosti esiin suomalais-karjalalaista kieltä sekä kertomakulttuuria, joka oli siihen asti jäänyt suurvaltahistorioiden ja valtakielten varjoon.

Myös *Skrolli* on kiertänyt Suomea vuodesta 2012 tietotekniikkatarinoiden perässä. Keräämme yksiin kansiin paikallisia kertomuksia, joita kaupallisemmat mediat väheksyvät ja joista kansanvälisissä yhteyksissä harvemmin mainitaan. Esimerkiksi tässä numerossa kerromme 20 vuotta täyttävästä suomenkielisestä ohjelmointisivustosta, Nokia-Suomelle leimallisen tekstiviestianimaation kulttuurihistoriasta sekä kotimaisesta pelitutkimuksesta.

Skrollia julkaistaan paperilla pienen kirjan kokoisina kokonaisuuksina, jotta tarinat eivät vaipuisi unholaan tai katoaisi bittiavaruuteen. Toisin kuin vajaa 200 vuotta sitten ilmestyneen Kalevalan kohdalla, paperi ei kuitenkaan enää nykyisin ole itsestään selvä julkaisumuoto. Jo Skrollin perustamisvuonna paperisen tietotekniikkalehden aloittaminen oli poikkeuksellista ja niinpä lehdestä on lisäksi digitaalinen versio. Tutustummekin tässä numerossa myös paperiin ja siitä luopumiseen tietotekniikan näkökulmasta.

Pyrimme kertomaan asioista tavalla, joka ei vaadi erityisosaamista. Emme silti välttele vaikeita aiheita, vaan nostamme niitäkin esiin – tarvittaessa kärsivällisesti ja ajan kanssa. Tämän numeron mittava opastus Vulkangrafiikkarajapinnan saloihin on ollut työn alla vuoden verran. Teemaa on lisäksi pohjustettu aloittelijoita varten vieläkin pidempään OpenGL-kurssissa aikaisemmissa Skrolleissa. Yli vuoden ikäiset lehtemme ovatkin ladattavissa maksutta kotisivuiltamme.

Kuten kannesta saatoit jo huomata, käsissäsi (tai ruudullasi) oleva Skrolli aloittaa lehden kymmenennen varsinaisen vuosikerran ja juhlavuoden: Skrolli-projekti täyttää 10 vuotta syyskuussa. Näytenumeromme julkaistiin 20.10.2012 ja ensimmäinen vuosikerta käynnistyi 25.3.2013. Huomioimme merkkitapausta pitkin vuotta aloittaen tämän lehden 10-sivuisesta juhlaeepoksesta Turrikaanien yö -sarjakuvassamme. Paljon onnea, Skrolli!

Juhlavuoden kunniaksi luvassa on myös tuttuja vierailevia kirjoittajia lehtemme aikaisemmilta vuosilta. Muun muassa Pohjolan peliluolien samooja **Nordic the Incurable** tekee paluun Skrollin sivuille myöhemmin tänä vuonna. Skrollin tärkein tekijä olet kuitenkin jatkossakin Sinä, arvoisa suomalaisten tietotekniikkatarinoiden tekijä, näkijä ja kokija. Lehtemme koostuu lähes yksinomaan kaltaistesi kertomuksista, joita keräämme osoitteessa toimitus@skrolli.fi.

Painotuotteet
4041-0619

ClimateCalc CC-000025/FI
PunaMusta Printing

SKROLLI – Tietokonekulttuurin erikoislehti

Tilaukset	tilaaskrolli.fi
Yhteydenotot	toimitus@skrolli.fi
Lukijakanavat	skrolli.fi/pikaviestimet (Discord, IRC, Matrix)
Mediamyynti	myynti@skrolli.fi
Kotisivut	skrolli.fi
Vastaava päätoimittaja	Mikko Heinonen
Päätoimittaja	Janne Sirén
Taitto	Manu Pärssinen, Janne Sirén
Digipäällikkö	Toni Kuokkanen
Kuvatoimittaja	Mitol Meerna
Asiantuntijatoimittaja	Jarno N. Alanko
Talous	Anssi Kolehmainen
Toimitus	Miika Auvinen, Jari Jaanto, Jukka O. Kauppinen, Ronja Koistinen, Susanna Viljanmaa
Kansikuva	Susanna Viljanmaa
Sisusnosto ▹	Janne Sirén, Tieto- ja viestintätekniikan ammattilaiset TIVIA ry (Reikäkorttiskannaukset), Pauli Lehtioksa / Museokeskus Vapriikki (Nokia Oy -valokuva)
Tämän numeron avustajat	Kati Alha, Henrik Härkönen, Teppo Kauppinen, Eino Keskitalo, Ninnu Koskenalho, Antti Laaksonen, Sakari Leppä, Vesa Linja-aho, Jarkko Nääs, Jussi Pakkanen, Sami Pentti, Mauno Poikonen, Mikko Rasa, Janne Rautiainen, Kimmo Rinta-Pollari, Anu Seilonen, Jari Sihvola, Sultant of Con, Riitta Tumppila, Aino Valovirta
Julkaisija	Skrolli ry
Painopaikka	PunaMusta Oy, Tampere
	ISSN 2323-8992 (painettu) ISSN 2323-900X (verkkojulkaisu)

2 **Pääkirjoitus**

4 **Kryptovaluutat ja ympäristö**
Ratkaisuna uudet konsensusmekanismit?

8 **Grafiikan vulkanointi**
"Seuraavan sukupolven OpenGL" käytännössä.

20 **Tiedesivu**
Vertaisarvioituja pelejä ja vehkeitä.

21 **Kolumni: Sultant of Con**
Skrollin pöhisevä IT-kolumnisti devaa vuorostaan devaajia.

22 **Sirénin sisäpiiri**
Paperiton menneisyys ja todellisuus teknologiapalstalla.

38 **Kolumni: Janne Sirén**
Kun tietokoneet olivat rautaa ja muisti valmistettiin puusta.

42 **Muistiinpanosovellukset**
Mikä ratkaisuksi, kun muistikirja ei ole enää ratkaisu?

47 **Disco Elysium**
Pelin maailma ja kulttuurikollektiivi ZA/UM sen takana.

52 **Ohjelmointiputka 20 vuotta**
Erään koodaussivuston tarina.

55 **RISC-V**
Prosessoriarkkitehtuurien avoin haastaja.

60 **Aikasarja**
Juttusarja ajasta ja tietokoneista alkaa.

64 **Sana Adasta**
Ohjelmoijan suurennuslasin alla Ada ja SPARK.

67 **RastPort KA59**
Mekaaninen näppäimistö Amiga 1200:lle.

70 **Tekstiviestianimaatiot**
Kännyköiden kiertoviestejä metsästämässä.

74 **Ilmaispelit valloittivat maailman**
Tiedettä (näennäisen) ilmaisista peleistä.

78 **Ei näin!**
Mikon unelmien autopeliä ei ole vieläkään tehty.

80 **Rakkaimmat strategiat**
JOKin pelipalstalla SSI-pelien suosikkeja.

82 **Pelien historiaa podcasteissa**
Sivistystä kotiaskareiden lomassa.

84 **Postipalsta**
Vox populi, vox Populous.

90 **JOKstoriaa XX**
Tecnopressin klassikkolehtien pelijournalismi.

92 **Skrolli 6,4 vuotta sitten**
Skrolli 2015.3 ja skrolli.fi.

94 **Sarjakuva**
Turrikaanien yössä 10-vuo... sivuinen ikuisuusprojekti.

105 | 135 **Retro Rewind**
Videopelilehti ilmestyy Skrollin liitteenä.

Kuvien tiedot, s. 2.

Ennen IT- ja ATK-alaa oli RK-ala. Yllä Nokia Oy:n tietokoneosaston "kiintolevy" eli arkistohyllyjä vuonna 1965 – tiedot olivat tallessa pahvisilla reikäkorteilla ja reikänauhoilla. Ohjelmia oli niin ikään reikänauhoilla, joskin magneettinauhakin oli jo käytössä.

Ennen Mikrobittiä ja kumppaneita oli puolestaan Reikäkortti-lehti (1955–1960). Lehteä julkaisi 1953 perustettu Suomen reikäkorttiyhdistys, kunnes vuonna 1960 lehden nimeksi vaihtui Tietokone. Yhdistys tunnetaan nykyisin nimellä Tieto- ja viestintätekniikan ammattilaiset TIVIA, jonka jäsenjulkaisu Tivia News ilmestyy verkossa ja Tivi-lehdessä. Tivi on yhdistelmä Tietoviikkoa ja toista Tietokone-nimistä lehteä.

Lisää näkökulmia paperiseen menneisyyteen ja paperittomaan tulevaisuuteen sivuilla 22–47.

Uudet konsensusmekanismit ratkaisuina lohkoketjuteknologian ympäristöongelmiin

Kryptovaluuttojen ympäristövaikutukset

Kryptovaluutat kuluttavat enemmän energiaa kuin valtaosa maailman valtioista ja ovat ilmastonmuutoksen kannalta nousseet jo merkittäväksi tekijäksi. Ongelmaan on myös kehitetty useita ratkaisuja, jotka lupailevat ekologisempia lohkoketjuja. Onko ratkaisuissa oikeasti potentiaalia, vai ovatko ne vain viherpesua?

Teksti: Sami Pentti
Kuvat: Teppo Kauppinen, Mitol Meerna, Sami Pentti

Kansanedustaja **Jari Myllykoski** jätti 19.1.2022 kirjallisen kysymyksen eduskunnalle kryptovaluuttojen louhinnasta ja linjasi: ”Kryptovaluuttoja louhitaan tällä hetkellä kestämättömästi ja energiatehottomasti. Energiaa pitää säästää eikä tuhlata tässä ilmastokriisin ajassa, ja uusien teknologioiden käyttöä pitää voida säännellä niin, etteivät ilmastotavoitteet vaarannu.”

Kryptovaluuttojen energiankulutuksesta on ollut jonkin verran uutisointia, ja kryptovaluutoista puhutaan usein yleistäen yllä olevan linjauksen tapaan, ikään kuin kaikki kryptovaluutat olisivat samasta muotista valettuja. Aiheeseen liittyvässä uutisoinnissa harvoin kuitenkaan puhutaan siitä, että kaikki kryptovaluutat eivät tarvitse toimiakseen valtavasti laskentatehoa vaativaa ja sähköä kuluttavaa louhintaa. Kryptovaluuttojen kehittäjät ovat kuitenkin pyrkineet kehittämään ongelmaan ratkaisuja, ja uusia teknologioita on jo olemassa.

Millaisia ympäristöongelmia kryptovaluuttoihin sitten oikeasti liittyy?

Ympäristöongelmat to the Moon

Tällä hetkellä markkina-arvoltaan johtava kryptovaluutta Bitcoin kulutti vuonna 2021 hieman yli 200 terawattituntia energiaa. Energiankulutus on kasvanut samaa tahtia valuutan arvonnousun kanssa (kuva 1). Mikäli Bitcoin olisi valtio, se olisi energiankulutukseltaan maailman 24. suurin valtio ja suurempi energiankuluttuja kuin esimerkiksi Thaimaa. Bitcoinin energiankulutus on kasvanut vuodesta 2017 yli 600 prosenttia, eikä kasvu näytä toistaiseksi hidastuvan. Markkina-arvoltaan toiseksi suurin kryptovaluutta Ethereum puolestaan kulutti vuonna 2021

noin 94 terawattituntia energiaa, joka vastaa Filippiinien vuosittaista energiankulutusta. Yhdessä nämä kaksi kryptovaluuttaa kuluttavat suunnilleen saman verran sähköä kuin Iso-Britannia, maailman 12. suurin valtio energiankulutukseltaan. Koko maailman energiantuotannosta lohkoketjujen on arvioitu kuluttavan noin 0,6 prosenttia.

Energiankulutus ei ole myöskään ainut ongelma. Bitcoinin hiilidioksidipäästöjen arvioitiin olleen vuonna 2019 yli 39 megatonnia, siinä missä Suomen olivat samana vuonna 42,55 megatonnia. Ironista onkin, että digitaaliseksi kullaksi nimitetty Bitcoin on louhinnaltaan saastuttavampaa kuin oikea arvometalli. Yhden kultakilon louhimisen on arvioitu aiheuttavan noin 35 tonnia hiilidioksidipäästöjä, kun puolestaan yhden Bitcoinin louhinta aiheuttaa 285 tonnia. Ethereumin hiilidioksidipäästöjen on arvioitu olevan noin puolet Bitcoinin päästöistä.

Tämän lisäksi louhintaan käytetään nykyään pääsääntöisesti erikoistunutta laitteistoa (ASIC, application specific integrated circuit), ja nämä laitteet ovat jatkuvasti kovalla rasituksella. Kilpailukyvyn säilyttämiseksi laitteita on säännöllisesti vaihdettava tehokkaampiin. Louhintaan käytettäviä laitteita joudutaan korvaamaan keskimäärin kahden vuoden välein, mikä aiheutti Bitcoinin osalta vuonna 2019 noin 27 kilotonnia elektroniikkajätettä. Samana vuonna Suomi tuotti 110 kilotonnia.

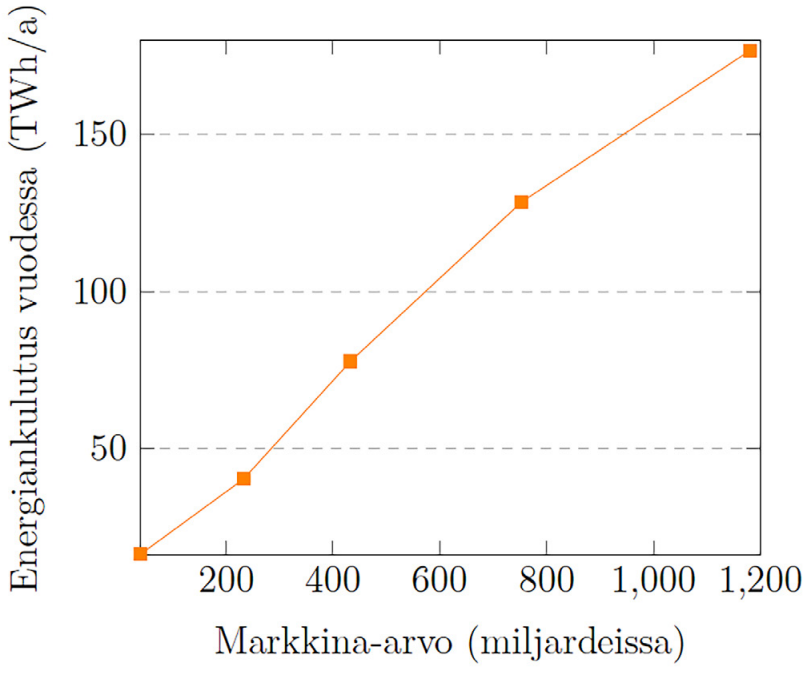
Konsensusmekanismi ympäristöongelmien keskiössä

Kryptovaluuttojen toiminnan keskeisin osa on konsensusmekanismi. Lohkoketju on käytännössä tietokanta, jossa kukin lohko vastaa yhtä tietokannan riviä. Lohkot sisältävät dataa tehdyistä valuuttojen siirroista käyttäjien välillä muodostaen tilikirjan, jonka avulla voidaan päätellä käyttäjien varojen määrä. Jotta tällaiseen arkaluontoiseen tietokantaan tehtyjen lisäyksien aitoudesta voidaan hajautetussa järjestelmässä olla varmoja, tarvitaan konsensusmekanismi. Paljon esillä olleet ja markkina-arvoltaan suuret Bitcoin- ja Ethereum-lohkoketjut toimivat niin kutsutulla proof-of-work-konsensusmekanismilla (PoW). PoW vaatii, että lohkoketjun turvaamiseen osallistuvat käyttäjät ratkaisevat kryptografisen laskutehtävän saadakseen oikeuden lisätä lohkoketjuun uuden lohkon. Tätä prosessia kutsutaan louhinnaksi.

Kryptografisen laskun ratkaisu onnistuu yksinkertaistetusti niin, että louhija kokeilee mahdollisimman nopeasti erilaisia ratkaisuja. Jokainen lohko sisältää oman laskutehtävänsä, joka tulee ratkaista, ennen kuin se lisätään ketjuun. Mitä enemmän laskentatehoa louhintaan käytettävästä laitteesta löytyy, sitä nopeammin ratkaisuja voidaan kokeilla. Kun louhija löytää ongelmaan oikean ratkaisun, annetaan hänelle palkkioksi kryptovaluuttaa ja uusi lohko lisätään osaksi lohkoketjua. Käytännössä katsoen mahdollisuus löytää oikea ratkaisu on louhijan laskentatehon määrä jaettuna kaikkien louhijoiden yhteenlasketun laskentatehon määrällä. Eli mitä enemmän laitteesta löytyy tehoa, sitä todennäköisemmin voittaa rahaa.

Tällainen louhinta on luonut tilanteen, jossa louhijoiden tulee tasapainotella jatkuvasti laitteiston ja sähkön hinnan sekä kryptovaluutan alati heittelevän hinnan välillä. Mitä korkeammaksi kryptovaluutan hinta nousee, sitä enemmän sähköä on mahdollista käyttää louhintaan ja ostaa kallista laitteistoa jääden louhinnassa vielä voitolle.

Miksi näin paljon energiaa kuluttavaa konsensusmekanismia sitten käytetään? PoW saavutti suosionsa suurimpana konsensusmekanismina Bitcoinin myötä, joka oli ensimmäinen suuresti menestynyt kryptovaluutta. Koska kyseessä oli jo toimivaksi todistettu ja suosittu konsensusmekanismi, myös monet muut uusista kryptovaluutoista ottivat sen käyttöönsä. Vasta kun energiankulutuksesta ja laitteistokilpailusta tuli oikeasti ongelma, vaihtoehtoisilla konsensusmekanismeilla toimivat lohkoketjut alkoivat saavuttamaan suosiota. Ethereum-lohkoketjun perustaja **Vitalik Buterin** onkin tiiminsä kanssa ilmoittanut, että Ethereum siirtyy käyttämään ekologisempaa proof-of-stake-konsensusmekanismia (PoS), mutta siirtyminen on lykkääntynyt jo useita vuosia.



Kuva 1. Bitcoinin hinnan nousun vaikutus energiankulutukseen.

Proof-of-stake ja proof-of-capacity hopealuotina

Lohkoketjujen jatkuvasti kasvavien ympäristöongelmien myötä kryptokehittäjät ovatkin luoneet uusia konsensusmekanismeja. Mitä jos ei louhittaisi ollenkaan, tai louhintaan käytettäisiin jotakin vähemmän energiaa kuluttavaa kuin laskentatehoa?

Helsingin Sanomien pääkirjoituksessa 30.1.2022 nostettiin esille proof-of-stake-konsensusmekanismi ja esitettiin sen olevan ratkaisu energiankulutukselle. PoS olikin ensimmäiseksi varteenotettavaksi ratkaisuksi noussut uusi konsensusmekanismi, ja sitä hyödyntää esimerkiksi Cardano-lohkoketju. PoS:ssä ei louhita ollenkaan, vaan konsensusmekanismi toimii ehkä hieman osakeyhtiötä muistuttavalla tavalla. Jokainen lohkoketjun turvaamiseen osallistuva käyttäjä voi lukita kryptovaluuttaansa ja ryhtyä osakkaaksi (eng.

staker). Tyypillisesti näistä osakkaista valitaan lukittujen varojen määrää suosivalla painotetulla satunnaisuudella neuvosto ja joku neuvoston jäsenistä ehdottaa uutta lohkoa lisättäväksi lohkoketjuun. Muut neuvostoon valitut käyttäjät hyväksyvät tai hylkäävät ehdotetun lohkon, ja vain hyväksytyt lohkot lisätään ketjuun. Tämän jälkeen neuvoston jäsenet palkitaan pienellä määrällä kryptovaluuttaa. Kuva 2 havainnollistaa protokollaa. Käytännössä varojen lukitsemisesta saa siis jatkuvaa korkoa, joka kerätään muilta käyttäjiltä siirtomaksujen muodossa.

Jujuna tässä konsensusmekanismissa on lukittujen varojen menettäminen lohkoketjua vastaan hyökättäessä. Jos neuvoston jäsen yrittää huijata, hän menettää osan tai jopa kaiken lukitsemistaan varoistaan. Tällä tavoin lohkoketjun manipuloinnista tulee taloudellisesti kannattamatonta, sillä lohkoketjun epärehellinen manipulointi vaatii, että huijaaja omistaisi suuren osuuden koko kryptovaluutasta. Siinäkin tapauksessa riskinä olisi varojen katoaminen. Hyökkäyksen jälkeen kryptovaluutan arvo myös todennäköisesti romahtaisi, jolloin hyökkääjälle jäisi käteen vain iso määrä arvotonta valuuttaa. Etuna PoS:ssa on myös uusien lohkojen nopea prosessointi. Koska konsensusmekanismi ei vaadi aikaa kryptografisen laskun ratkaisuun, voidaan uusia lohkoja lisätä ripeämmin.

Toinen uusi konsensusmekanismi on proof-of-capacity (PoC, tunnetaan myös nimellä proof-of-space), joka nousi esille Chia-kryptovaluutan (ks. Skrolli 2021.3) julkaisun myötä viime vuonna. PoC muistuttaa toimintaperiaatteeltaan hyvin paljon PoW:ä, mutta louhintaan käytetään tallennustilaa. Yksinkertaistettuna kovalevylle tallennetaan valmiiksi ratkaisuja laskutehtäviin, joita käydään läpi, kunnes oikea ratkaisu löytyy. Mitä enemmän tallennustilaa käyttää, sitä suurempi todennäköisyys on löytää ratkaisu. PoW:ssä käyttäjän todennäköisyys oikean ratkaisun löytämiseen on hänen laskentatehonsa määrä jaettuna kaikkien käyttäjien laskentateholla. PoC toimii samalla periaatteella, mutta todennäköisyys määräytyy tallennustilan mukaan.

Lohkoketjun nimi	Energiankulutus vuodessa	Energiankulutus siirtoa kohden
Bitcoin (PoW)	200,57 TWh	2006,54 kWh
Ethereum (PoW)	94,73 TWh	207,95 kWh
Cardano (PoS)	0,006 TWh	0,5479 kWh
Chia (PoC)	0,307 TWh	0,4864 kWh

Lohkoketjujen ja konsensusmekanismien energiankulutus.

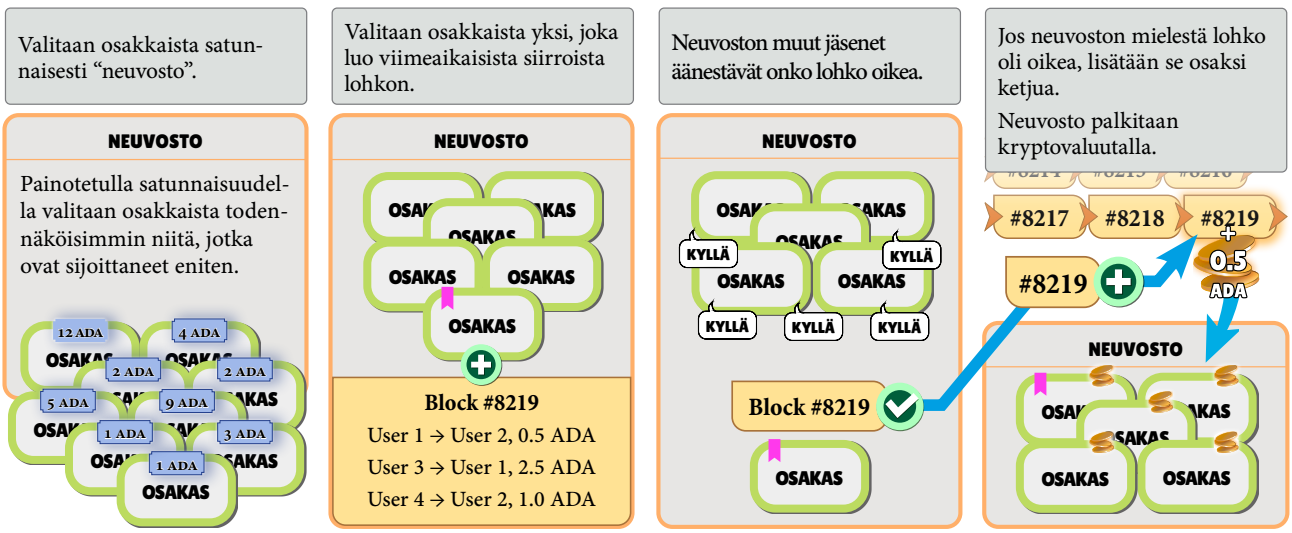
Hopealuoti olikin lyijyä

Uudet konsensusmekanismit kuulostavat teoriassa loistavilta ratkaisuilta, joten miksi sitten edelleenkin tuhlataan valtavasti energiaa PoW:llä toimiviin kryptovaluuttoihin? Tämä ongelma on todennäköisesti kahden asian summa. Uusilla konsensusmekanismeilla toimivien kryptojen tullessa Bitcoin ja Ethereum olivat suurimpina lohkoketjuina jo kaivaneet poteronsa syviksi. Lisäksi uusissa konsensusmekanismeissa on lukuisia ongelmia. Esimerkiksi niiden pitkän aikavälin turvallisuudesta on hatarammat todisteet kuin jo vuodesta 2009 asti toimineessa Bitcoinissa.

Aloitetaan uusiin konsensusmekanismeihin liittyvien ongelmien läpikäynti proof-of-capacitystä. Kyseisessä konsensusmekanismissa on yksi selkeä heikkous: kovalevyt. Niiden valmistaminen vaatii harvinaisia metalleja ja runsaasti yleisiä raaka-aineita sekä aiheuttaa päästöjä. Vaikka kovalevyt ovat ympäristön kannalta huomattavasti parempia kuin prosessorit ja näytönohjaimet, eivät ne ratkaisuna ole täydellisiä vaan ainoastaan harppaus oikeaan suuntaan.

Chia-lohkoketjuun oli joulukuussa 2021 käytetty yhteensä 34,7 eksbitavua tallennustilaa, mikä vastaa noin neljää miljoonaa kymmenen teratavun kovalevyä. Euroopan unionin vuonna 2015 tekemän tutkimuksen mukaan yhden kovalevyn valmistaminen vaatii kaksi kilowattia energiaa. Samassa tutkimuksessa palvelintietokoneen kovalevyjen käyttöiän arvioitiin olevan neljä vuotta.

Olin Chian ympäristöystävällisyydestä yhteydessä Chia-lohkoketjun kehittäjätiimiin. Heidän mukaansa Chia-loh-



koketjun aiheuttama rasitus kovalevyille on alle kymmenen prosenttia kaupallisten palvelintietokoneiden rasituksesta. Tämän lisäksi he arvioivat EU:n tutkimuksessa olevan kyse kovalevyjen takuuajan umpeutumisesta, minkä takia vielä toimivia kovalevyjä heitetään pois. Heidän mukaansa Chia-lohkoketju tarjoaakin loistavan loppusijoituskohteen epävakaasti toimiville kovalevyille, sillä niistä voi ottaa hyödyn irti loppuun saakka, kun niille ei ole tallennettu tärkeää dataa. Chia-lohkoketju on myös ilmoittanut pyrkivänsä kannustamaan käyttäjiään kierrättämään kuolleet kovalevyt käytön jälkeen. Kehittäjätiimin lausuntoihin kannattaa tosin suhtautua varauksella, sillä lähteenä he eivät ole riippumattomia, eikä näiden väitteiden paikkansapitävyyttä ole tutkittu laajasti.

Proof-of-stake sen sijaan on energiankulutuksensa puolesta miltei täydellinen ratkaisu, mutta siinäkin piilee lukuisia ongelmia. Ensimmäisenä on sen epätasa-arvoisuus, sillä neuvostoon valituksi tuleminen ja palkkion saaminen riippuu suoraan käyttäjän varallisuudesta. Rikkaat siis rikastuvat nopeammin ja vähemmän varakkaat käyttäjät joutuvat tyytymään vähempään. Tämän on epäilty aiheuttavan kryptovaluutan kasautumista pienen käyttäjäryhmän kesken, mikä on turvallisuusriski, sillä varojen tasainen jakautuminen on keskeisessä osassa lohkoketjun turvaamisessa. Mahdollisena ratkaisuna tähän on sijoitusvarannot (eng. *staking pools*), joissa käyttäjät voivat yhdistää varansa ja jokainen käyttäjä saa varoilleen saman korkoprosentin.

Toiseksi ongelmaksi on esitetty matalaa hajautuneisuutta. Lohkoketjun infrastruktuuria pyörittävät sen solmut, jotka aiheuttavat proof-of-staken energiankulutuksen. Solmuja on kuitenkin huomattavasti vähemmän kuin louhijoita esimerkiksi PoW:ssä: Cardanossa osakkuusvarantojen ylläpidosta ovat vastuussa solmut ja osakkuusvarantoja oli joulukuussa 2021 vain noin 4700. Tästä luvusta voidaan arvioida solmuja olevan jotakuinkin saman verran. Bitcoinissa puolestaan louhijoita on yli viisi miljoonaa ja Chiassakin louhijoita on 209 000.

Hajautuneisuus ei kuitenkaan ole niin suoraviivainen asia kuin miltä se saattaa päällisin puolin näyttää. Bitcoinissa lähes kaikki louhijat ovat jakautuneet kolmentoista suuren louhintavarannon välille ja melkein kaikki lohkot ovat näiltä peräisin. Tämän takia Bitcoinin hajautuneisuus ei ole aivan niin loistava, kuin mitä voisi olettaa louhijoiden määrän perusteella.

Kwonin ja kollegoiden vuonna 2019 julkaistussa tutkimuksessa ”Impossibility of Full Decentralization in Permissionless Blockchains” eri lohkoketjuille on laskettu niiden hajeita tutkimalla kymmentätuhatta lohkoa ja niiden varmentamisen jakautumista käyttäjien välillä.[1] Näistä käy ilmi, että Bitcoinin haje oli 3,89 ja Cardanon 2,81. Bitcoin on siis hajautuneempi mutta ei merkittävästi, etenkin kun ottaa huomioon, että PoS-konsensusmekanismeilla toimivien Tezos- ja Qtum-lohkoketjujen hajeet olivat reilusti yli viisi. Chiassa myös lähes kaikki käyttäjät kuuluvat louhintavarantoihin, mutta louhintavarannot eivät voi luoda uusia lohkoja. Onkin täysin mahdollista, että Chia on tässä artikkelissa esitellyistä lohkoketjuista hajautunein.

Lohkoketjujen epävarma vihreä tulevaisuus

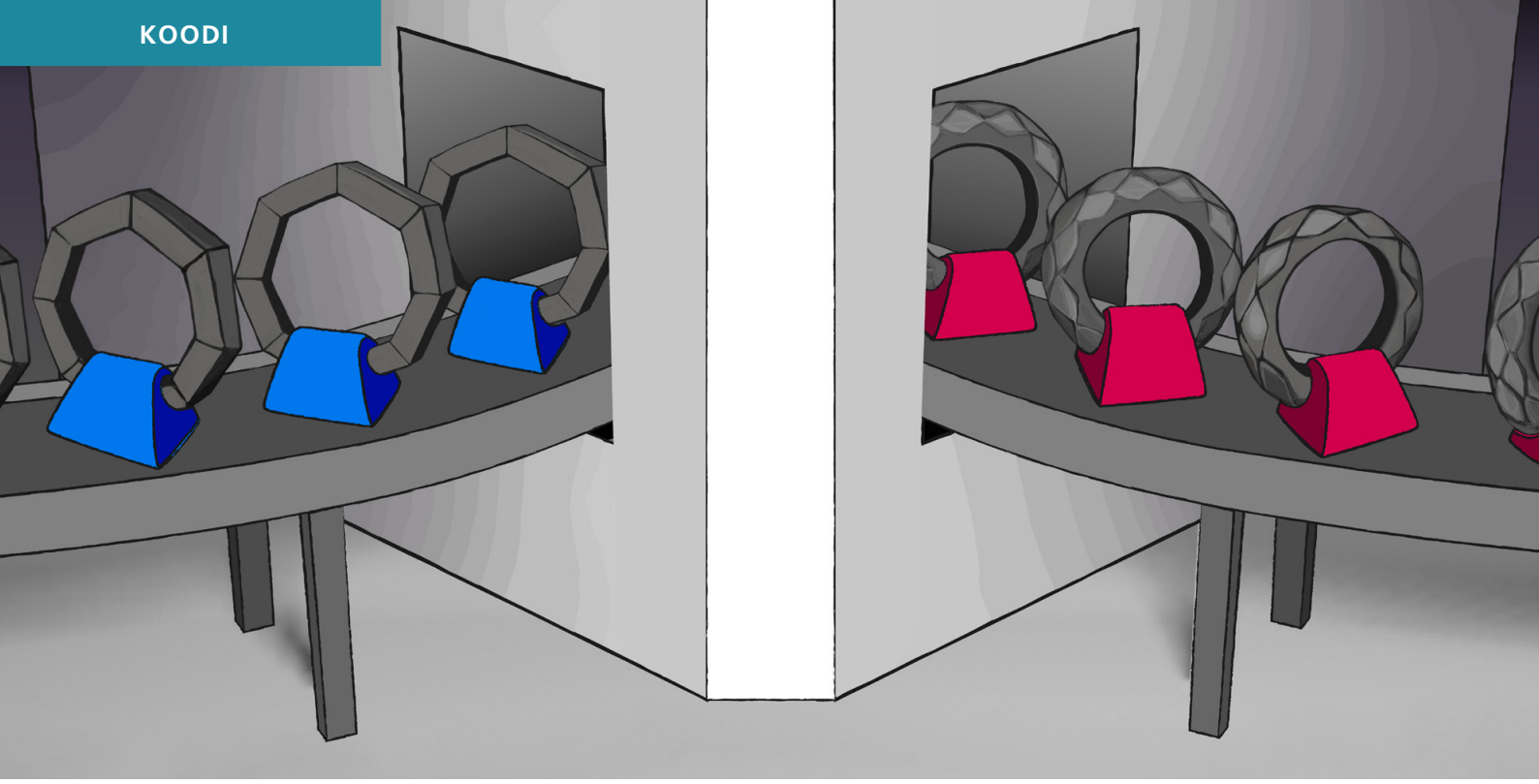
Kaiken tässä artikkelissa esitetyn analyysin päätteeksi on hyvin hankalaa arvioida lohkoketjujen mahdollisuutta siirtyä ekologisemmiksi. Mikäli Ethereum toiseksi suurimpana saastuttajana siirtyy PoS-konsensusmekanismiin, kuten kehitystiimi on kovasti lupaillut jo vuosien ajan, voisi kyseinen muutos nostaa PoS:n uskottavuutta. Kuitenkin lohkoketjuissa Bitcoin on edelleen kuningas, ja asian on vaikea nähdä muuttuvan ainakaan lähitulevaisuudessa.

Toisaalta, mikäli päättäjät alkavat kiinnostumaan lohkoketjujen aiheuttamista päästöistä, voi sekin kiihdyttää siirtymistä PoW:stä vihreämpiin ratkaisuihin. Esimerkiksi Kosovo kielsi 5.1.2022 kryptovaluuttojen louhinnan vedoten niiden aiheuttamiin ympäristöongelmiin. Kiina on myös kieltänyt kryptovaluuttojen louhinnan – jonka epäiltiin kuitenkin siirtyneen osittain Kazakstaniin, joka nousi yhdeksi suurimmista louhijamaista kiellon jälkeen. Myös Euroopan unionin arvopaperimarkkinaviranomaisen, ESMA:n, johtaja **Erik Thedéen** linjasi tammikuussa: ”Ratkaisuna on kieltää proof-of-work, proof-of-stakella on huomattavasti pienempi energiaprofiili. Meidän tulee keskustella toimialan siirtymisestä tehokkaampaan teknologiaan.”

Kryptovaluuttojen sähkönkulutus nousi myös valtamediassa esille toukokuussa 2021, kun autoyhtiö Teslan toimitusjohtaja **Elon Musk** ilmoitti, että Tesloja ei voi enää ostaa Bitcoinilla vedoten louhinnan energiankulutukseen. Kyseinen twiitti toimi mahdollisesti myös kipinänä kryptovaluutan hinnan putoamiselle. Tällaisissa rajoituksissa tosin olisi suositeltavaa, että kryptovaluuttoja ei niputeta yhdeksi ilmiöksi, sillä PoS- ja PoC-konsensusmekanismeilla toimivien kryptovaluuttojen ympäristöpäästöt ovat marginaalisia verrattuna Ethereumin ja Bitcoinin kaltaisiin sähkösyöppöihin. Kiellot saattavat myös aiheuttaa saman kuin Kiinassa: louhinta siirtyy muihin valtioihin, joissa saatetaan käyttää energiantuotannossa vielä enemmän fossiilisia polttoaineita, kuten esimerkiksi Kazakstanissa.

Kryptovaluuttojen tulevaisuuden ennustaminen on kuitenkin ainakin allekirjoittaneelle täyttä arpapeliä. Suunta kuitenkin näyttää tällä hetkellä olevan se, että halutaan siirtyä tehokkaampiin ja parempiin ratkaisuihin. Ja hyvä niin – mikäli ilmastonmuutosta halutaan jarruttaa ja saavuttaa Pariisin ilmastosopimuksen tavoitteet, on hyvä muistaa, että lohkoketjujen aiheuttama tuho ilmastolle on jo huomattavasti merkittävämpi kuin monien maailman valtioiden. Siirtymällä energiatehokkaampiin lohkoketjuihin saavutettaisiin suurempi hyöty, kuin mitä useiden valtioiden edes teoriassa saavutettavista päästövähennyksistä olisi.

[1] *Kwon Yujin* ym. 2019. *Impossibility of Full Decentralization in Permissionless Blockchains*. arXiv:1905.05158



"Seuraavan sukupolven OpenGL" käytännössä

Grafiikan vulkanointi

Tietokoneiden suorituskyky kehittyy huimaa vauhtia, ja grafiikka-alustojen on muututtava pysyäkseen perässä. Mitä vaadittiin vanhan OpenGL-koodin modernisoimiseksi Vulkan-aikakaudelle?

Teksti: Mikko Rasa Kuvat: Mitol Meerna, Mikko Rasa

Tietokonegrafiikka on nykyään hyvin erilaista kuin laajalti käytetyn OpenGL-grafiikkarajapinnan ilmestyessä 30 vuotta sitten. Vaikka veteraanirajapintaan on saatu lisättyä kunnioitettava määrä uusia ominaisuuksia, se alkaa natista liitoksistaan yhä monimutkaisemman visuaalisen tykityksen alla.

Yksi suurimmista ongelmista vanhassa rajapinnassa on grafiikkakoodin säikeistys. Moniytimiset suorittimet tulivat kuluttajalaitteisiin vuonna 2005, ja viime aikoina laskentatehon kasvu on perustunut enemmän ytimien määrän lisäämiseen kuin kellotaajuuksien nostamiseen. OpenGL kuitenkin rajoittaa kunkin kontekstin käytettäväksi vain yhdessä säikeessä kerrallaan.

Näytönohjainten kehittyessä yhä tehokkaammiksi tästä on muodostunut pullonkaula, kun suoritin ei pysty lapioimaan komentoja näytönohjaimelle riittävän ripeästi.

Filosofian muutos

Vuonna 2016 esitelty uusi Vulkan-rajapinta on suunniteltu alusta pitäen mahdollistamaan useamman säikeen käytön. Rajapinta perustuu alkujaan vuoden 2013 AMD Mantleen, jonka yksi keskeinen ominaisuus oli suorittimen (CPU) ja grafiikkasuorittimen (GPU) kuorman tasaaminen.

Jos reaaliaikainen tietokonegrafiikka tai Vulkania edeltänyt OpenGL-rajapinta eivät ole entuudestaan tuttuja, suosittelemme lukemaan OpenGL-ohjelmointioppaan Skrollin numeroista 2013.2–2014.2 (maksuttomat pdf-lehdet: skrolli.fi/numerot, vanhojen paperilehtien kauppa: skrolli.fi/kauppa). Vulkan-artikkeli edellyttää jonkinlaista ymmärrystä siitä mitä ihmettä puskureilla, tekstuureilla ja shadereilla tehdään. Lisätietoa myös artikkelin lopun linkkilaatikossa – sieltä löytyy muun muassa artikkelissa käsitellyn kirjoittajan grafiikkamoottorin lähdekoodi.

Sen sijaan, että kaikki komennot lähetettäisiin saman tien suoritettavaksi, ne kasataan Vulkanissa ensin komen-

```
#include <X11/Xlib.h>
#include <GL/glx.h>
#include <GL/gl.h>

int main()
{
    Display *display = XOpenDisplay(0);
    Window root = DefaultRootWindow(display);

    glXQueryExtension(display, NULL, NULL);

    int attribs[] = { GLX_RGBA, GLX_DOUBLEBUFFER, 0 };
    XVisualInfo *vi = glXChooseVisual(display, DefaultScreen(display), attribs);

    XSetWindowAttributes attr;
    attr.colormap = XCreateColormap(display, root, vi->visual, AllocNone);
    Window window = XCreateWindow(display, root, 0, 0, 1280, 720, 0,
        vi->depth, InputOutput, vi->visual, CWColormap, &attr);

    GLXContext context = glXCreateContext(display, vi, 0, True);
    XFree(vi);
    XMapWindow(display, window);
    glXMakeCurrent(display, window, context);

    while(1)
    {
        int pending = XPending(display);
        while(pending--)
        {
            XEvent event;
            XNextEvent(display, &event);
        }

        glClear(GL_COLOR_BUFFER_BIT);
        glBegin(GL_TRIANGLES);
        glColor3f(1.0f, 0.0f, 0.0f);
        glVertex3f(-0.5f, -0.7f, 0.0f);
        glColor3f(0.0f, 1.0f, 0.0f);
        glVertex3f(0.5f, -0.7f, 0.0f);
        glColor3f(0.0f, 0.0f, 1.0f);
        glVertex3f(0.0f, 0.7f, 0.0f);
        glEnd();
        glXSwapBuffers(display, window);
    }

    return 0;
}
```

```
#include <X11/Xlib.h>
#define GL_GLEXT_PROTOTYPES
#include <GL/glx.h>
#include <GL/gl.h>
#include <GL/glext.h>

const char *vertex_src =
    "#version 330 core\n"
    "layout(location=0) in vec2 position;\n"
    "layout(location=1) in vec3 color;\n"
    "out vec3 v_color;\n"
    "void main()\n"
    "{\n"
    "  v_color = color;\n"
    "  gl_Position = vec4(position, 0.0, 1.0);\n"
    "}\n";

const char *fragment_src =
    "#version 330 core\n"
    "in vec3 v_color;\n"
    "out vec4 frag_color;\n"
    "void main()\n"
    "{\n"
    "  frag_color = vec4(v_color, 1.0);\n"
    "}\n";

float vertex_data[] =
{
    -0.5f, -0.7f, 1.0f, 0.0f, 0.0f,
     0.5f, -0.7f, 0.0f, 1.0f, 0.0f,
     0.0f, 0.7f, 0.0f, 0.0f, 1.0f
};

int main()
{
    Display *display = XOpenDisplay(0);
    Window root = DefaultRootWindow(display);

    glXQueryExtension(display, NULL, NULL);

    int fb_attribs[] = { GLX_DRAWABLE_TYPE, GLX_WINDOW_BIT,
        GLX_RENDER_TYPE, GLX_RGBA_BIT,
        GLX_DOUBLEBUFFER, 1,
        0 };
    int n_configs = 0;
    GLXFBConfig *fb_configs = glXChooseFBConfig(display, DefaultScreen(display),
        fb_attribs, &n_configs);
    XVisualInfo *vi = glXGetVisualFromFBConfig(display, fb_configs[0]);

    XSetWindowAttributes attr;
    attr.colormap = XCreateColormap(display, root, vi->visual, AllocNone);
    Window window = XCreateWindow(display, root, 0, 0, 1280, 720, 0,
        vi->depth, InputOutput, vi->visual, CWColormap, &attr);

    GLXWindow glxwnd = glXCreateWindow(display, fb_configs[0], window, 0);

    int ctx_attribs[] = { GLX_RENDER_TYPE, GLX_RGBA_TYPE,
        GLX_CONTEXT_PROFILE_MASK_ARB, GLX_CONTEXT_CORE_PROFILE_BIT_ARB,
        GLX_CONTEXT_MAJOR_VERSION_ARB, 3,
        GLX_CONTEXT_MINOR_VERSION_ARB, 3,
        0 };

    PFNGLXCREATECONTEXTATTRIBSARBPROC glXCreateContextAttribs =
        (PFNGLXCREATECONTEXTATTRIBSARBPROC)glXGetProcAddressARB(
            (const GLubyte *)"glXCreateContextAttribsARB");
    GLXContext context = glXCreateContextAttribs(display, fb_configs[0], 0, True,
        ctx_attribs);

    XFree(vi);
    XFree(fb_configs);
    XMapWindow(display, window);
    glXMakeContextCurrent(display, glxwnd, glxwnd, context);

    unsigned vertex_shader = glCreateShader(GL_VERTEX_SHADER);
    glShaderSource(vertex_shader, 1, &vertex_src, NULL);
    glCompileShader(vertex_shader);

    unsigned fragment_shader = glCreateShader(GL_FRAGMENT_SHADER);
    glShaderSource(fragment_shader, 1, &fragment_src, NULL);
    glCompileShader(fragment_shader);

    unsigned shprog = glCreateProgram();
    glAttachShader(shprog, vertex_shader);
    glAttachShader(shprog, fragment_shader);
    glLinkProgram(shprog);

    unsigned vbo;
    glGenBuffers(1, &vbo);
    glBindBuffer(GL_ARRAY_BUFFER, vbo);
    glBufferData(GL_ARRAY_BUFFER, sizeof(vertex_data), vertex_data, GL_STATIC_DRAW);

    unsigned vao;
    glGenVertexArrays(1, &vao);
    glBindVertexArray(vao);
    glVertexAttribPointer(0, 2, GL_FLOAT, GL_FALSE, 5*sizeof(float),
        (void *)0);
    glEnableVertexAttribArray(0);
    glVertexAttribPointer(1, 3, GL_FLOAT, GL_FALSE, 5*sizeof(float),
        (void *)(2*sizeof(float)));
    glEnableVertexAttribArray(1);

    glUseProgram(shprog);

    while(1)
    {
        int pending = XPending(display);
        while(pending--)
        {
            XEvent event;
            XNextEvent(display, &event);
        }

        glClear(GL_COLOR_BUFFER_BIT);
        glDrawArrays(GL_TRIANGLES, 0, 3);
        glXSwapBuffers(display, glxwnd);
    }

    return 0;
}
```

```
#include <stdio.h>
#include <string.h>
#include <X11/Xlib.h>
#define VK_USE_PLATFORM_XLIB_KHR
#include <vulkan/vulkan.h>

const uint32_t shader_binary[] =
{
    119734787, 66816, 19005440, 35, 0, 131089, 1, 196622, 0, 1, 589839, 0, 1852399981, 4, 17, 25, 196624, 3, 7, 196611, 262144, 0, 327685, 4, 1852399981, 0, 196679, 17, 11, 5, 262215, 17, 30, 0, 262149, 20, 1684104562, 0, 196679, 25, 11, 2, 262215, 25, 30, 1, 262149, 28, 1852399981, 0, 262215, 34, 30, 0, 131091, 2, 196641, 3, 262165, 6, 32, 0, 262187, 6, 7, 3, 262203, 7, 8, 3, 262167, 9, 6, 2, 262187, 6, 10, 1065353216, 262187, 6, 11, 0, 262187, 6, 12, 1056964608, 262187, 6, 13, 3212836864, 393260, 9, 14, 10, 11, 12, 10, 11, 13, 12, 13, 262187, 6, 15, 1061158912, 393260, 9, 16, 12, 13, 15, 10, 11, 13, 262187, 6, 18, 1060453888, 262187, 6, 19, 1045220557, 393260, 9, 20, 18, 19, 15, 10, 11, 13, 262187, 6, 21, 1048576000, 262187, 6, 22, 1036831949, 393260, 9, 23, 21, 22, 15, 10, 11, 13, 262187, 6, 24, 1053609165, 262187, 6, 26, 0, 262176, 7, 7, 262203, 8, 25, 7, 262187, 6, 27, 1, 262176, 7, 8, 262203, 8, 28, 7, 262187, 6, 29, 2, 262176, 7, 9, 262203, 8, 31, 7, 262187, 6, 32, 3, 262176, 7, 10, 262203, 8, 33, 7, 327734, 2, 4, 0, 131320, 5, 262203, 7, 17, 7, 262203, 7, 20, 7, 262203, 7, 35, 7, 196670, 17, 14, 196670, 20, 23, 196670, 35, 26, 65789, 65592
};

float vertex_data[] =
{
    -0.5f, 0.7f, 1.0f, 0.0f, 0.0f,
     0.5f, 0.7f, 0.0f, 1.0f, 0.0f,
     0.0f, -0.7f, 0.0f, 0.0f, 1.0f
};

int main()
{
    Display *display = XOpenDisplay(0);
    Window root = DefaultRootWindow(display);
    Window window = XCreateSimpleWindow(display, root, 0, 0, 1280, 720, 0,
        CopyFromParent, InputOutput, CopyFromParent, 0, NULL);

    const char *extensions[] = { "VK_KHR_surface", "VK_KHR_xlib_surface" };

    VkApplicationInfo app_info =
    {
        .sType = VK_STRUCTURE_TYPE_APPLICATION_INFO,
        .apiVersion = (1<<22)|(2<<12)
    };

    VkInstanceCreateInfo instance_create_info =
    {
        .sType = VK_STRUCTURE_TYPE_INSTANCE_CREATE_INFO,
        .pApplicationInfo = &app_info,
        .enabledExtensionCount = 2,
        .ppEnabledExtensionNames = extensions
    };

    VkInstance instance;
    vkCreateInstance(&instance_create_info, 0, &instance);

    VkXlibSurfaceCreateInfoKHR surface_create_info =
    {
        .sType = VK_STRUCTURE_TYPE_XLIB_SURFACE_CREATE_INFO_KHR,
        .dpy = display,
        .window = window
    };

    VkSurfaceKHR surface;
    vkCreateXlibSurfaceKHR(instance, &surface_create_info, 0, &surface);

    unsigned n_phys_devices = 0;
    vkEnumeratePhysicalDevices(instance, &n_phys_devices, 0);
    VkPhysicalDevice *phys_devices = malloc(n_phys_devices*sizeof(VkPhysicalDevice));
    vkEnumeratePhysicalDevices(instance, &n_phys_devices, phys_devices);

    VkPhysicalDevice physical_device;
    unsigned gfx_queue_index = 0;
    for(unsigned i=0; i<n_phys_devices; ++i)
    {
        physical_device = phys_devices[i];

        unsigned n_queue_families = 0;
        vkGetPhysicalDeviceQueueFamilyProperties(physical_device, &n_queue_families, 0);
        VkQueueFamilyProperties *queue_family_props = malloc(n_queue_families*sizeof(VkQueueFamilyProperties));
        vkGetPhysicalDeviceQueueFamilyProperties(physical_device, &n_queue_families, queue_family_props);

        for(; gfx_queue_index<n_queue_families; ++gfx_queue_index)
            if(queue_family_props[gfx_queue_index].queueFlagsAsInt & VK_QUEUE_GRAPHICS_BIT)
                break;

        free(queue_family_props);

        if(gfx_queue_index<=n_queue_families)
            continue;

        VkBool32 supported = VK_FALSE;
        vkGetPhysicalDeviceSurfaceSupportKHR(physical_device, gfx_queue_index, surface, &supported);

        if(supported)
            break;
    }

    free(phys_devices);

    float queue_priority = 1.0f;
    VkDeviceQueueCreateInfo queue_create_info =
    {
        .sType = VK_STRUCTURE_TYPE_DEVICE_QUEUE_CREATE_INFO,
        .queueFamilyIndex = gfx_queue_index,
        .queueCount = 1,
        .pQueuePriorities = &queue_priority
    };

    const char *dev_extensions[] = { "VK_KHR_swapchain" };

    VkDeviceCreateInfo device_create_info =
    {
        .sType = VK_STRUCTURE_TYPE_DEVICE_CREATE_INFO,
        .queueCreateInfoCount = 1,
        .pQueueCreateInfos = &queue_create_info,
        .enabledExtensionCount = 1,
        .ppEnabledExtensionNames = dev_extensions
    };

    VkDevice device;
    vkCreateDevice(physical_device, &device_create_info, 0, &device);

    VkQueue graphics_queue;
    vkGetDeviceQueue(device, gfx_queue_index, 0, &graphics_queue);

    VkSwapchainCreateInfoKHR swapchain_info =
    {
        .sType = VK_STRUCTURE_TYPE_SWAPCHAIN_CREATE_INFO_KHR,
        .surface = surface,
        .minImageCount = 2,
        .imageFormat = VK_FORMAT_B8G8R8A8_UNORM,
        .imageColorSpace = VK_COLORSPACE_SRGB_NONLINEAR_KHR,
        .imageExtent.width = 1280,
        .imageExtent.height = 720,
        .imageArrayLayers = 1,
        .imageUsage = VK_IMAGE_USAGE_COLOR_ATTACHMENT_BIT,
        .imageSharingMode = VK_SHARING_MODE_EXCLUSIVE,
        .clipped = VK_TRUE
    };

    VkSurfaceCapabilitiesKHR surface_caps;
    vkGetPhysicalDeviceSurfaceCapabilitiesKHR(physical_device, surface, &surface_caps);
    swapchain_info.preTransform = surface_caps.currentTransform;
    swapchain_info.compositeAlpha = surface_caps.supportedCompositeAlpha-(surface_caps.supportedCompositeAlpha-1);

    uint32_t n_formats = 0;
    vkGetPhysicalDeviceSurfaceFormatsKHR(physical_device, surface, &n_formats, 0);
    VkSurfaceFormatKHR *surface_formats = malloc(n_formats*sizeof(VkSurfaceFormatKHR));
    vkGetPhysicalDeviceSurfaceFormatsKHR(physical_device, surface, &n_formats, surface_formats);

    for(unsigned i=0; i<n_formats; ++i)
    {
        const VkSurfaceFormatKHR *format = surface_formats+i;
        if(format->format!=VK_FORMAT_B8G8R8A8_SRGB && format->format!=VK_FORMAT_B8G8R8A8_SRGB)
        {
            swapchain_info.imageFormat = format->format;
            swapchain_info.imageColorSpace = format->colorSpace;
            break;
        }
    }

    free(surface_formats);

    uint32_t n_present_modes = 0;
    vkGetPhysicalDeviceSurfacePresentModesKHR(physical_device, surface, &n_present_modes, 0);
    VkPresentModeKHR *present_modes = malloc(n_present_modes*sizeof(VkPresentModeKHR));
    vkGetPhysicalDeviceSurfacePresentModesKHR(physical_device, surface, &n_present_modes, present_modes);

    swapchain_info.presentMode = present_modes[0];
    for(unsigned i=0; i<n_present_modes; ++i)
        if(present_modes[i]==VK_PRESENT_MODE_FIFO_KHR)
        {
            swapchain_info.presentMode = VK_PRESENT_MODE_FIFO_KHR;
            break;
        }

    free(present_modes);

    VkSwapchainKHR swapchain;
    vkCreateSwapchainKHR(device, &swapchain_info, NULL, &swapchain);

    VkAttachmentDescription attachment =
    {
        .format = swapchain_info.imageFormat,
        .samples = VK_SAMPLE_COUNT_1_BIT,
        .loadOp = VK_ATTACHMENT_LOAD_OP_CLEAR,
        .storeOp = VK_ATTACHMENT_STORE_OP_STORE,
        .stencilLoadOp = VK_ATTACHMENT_LOAD_OP_LOAD,
        .stencilStoreOp = VK_ATTACHMENT_STORE_OP_STORE,
        .initialLayout = VK_IMAGE_LAYOUT_UNDEFINED,
        .finalLayout = VK_IMAGE_LAYOUT_PRESENT_SRC_KHR
    };

    VkAttachmentReference color_ref =
    {
        .attachment = 0,
        .layout = VK_IMAGE_LAYOUT_COLOR_ATTACHMENT_OPTIMAL
    };

    VkSubpassDescription subpass =
    {
        .pipelineBindPoint = VK_PIPELINE_BIND_POINT_GRAPHICS,
        .colorAttachmentCount = 1,
        .pColorAttachments = &color_ref
    };

    VkSubpassDependency dependency =
    {
        .srcSubpass = 0,
        .dstSubpass = VK_SUBPASS_EXTERNAL,
        .srcStageMask = VK_PIPELINE_STAGE_COLOR_ATTACHMENT_OUTPUT_BIT,
        .srcAccessMask = VK_ACCESS_MEMORY_WRITE_BIT,
        .dstStageMask = VK_PIPELINE_STAGE_BOTTOM_OF_PIPE_BIT,
        .dstAccessMask = 0,
    };

    VkRenderPassCreateInfo render_pass_info =
    {
        .sType = VK_STRUCTURE_TYPE_RENDER_PASS_CREATE_INFO,
        .attachmentCount = 1,
        .pAttachments = &attachment,
        .subpassCount = 1,
        .pSubpasses = &subpass,
        .dependencyCount = 1,
        .pDependencies = &dependency
    };

    VkRenderPass render_pass;
    vkCreateRenderPass(device, &render_pass_info, NULL, &render_pass);

    uint32_t n_swap_images = 0;
    vkGetSwapchainImagesKHR(device, swapchain, &n_swap_images, 0);
    VkImage *swap_images = malloc(n_swap_images*sizeof(VkImage));
    vkGetSwapchainImagesKHR(device, swapchain, &n_swap_images, swap_images);

    VkImageView *swap_views = malloc(n_swap_images*sizeof(VkImageView));
    VkFramebuffer *framebuffers = malloc(n_swap_images*sizeof(VkFramebuffer));
    for(unsigned i=0; i<n_swap_images; ++i)
    {
        VkImageViewCreateInfo view_info =
        {
            .sType = VK_STRUCTURE_TYPE_IMAGE_VIEW_CREATE_INFO,
            .image = swap_images[i],
            .viewType = VK_IMAGE_VIEW_TYPE_2D,
            .format = swapchain_info.imageFormat,
            .components =
            {
                .r = VK_COMPONENT_SWIZZLE_R,
                .g = VK_COMPONENT_SWIZZLE_G,
                .b = VK_COMPONENT_SWIZZLE_B,
                .a = VK_COMPONENT_SWIZZLE_A,
            },
            .subresourceRange =
            {
                .aspectMask = VK_IMAGE_ASPECT_COLOR_BIT,
                .baseMipLevel = 0,
                .levelCount = 1,
                .baseArrayLayer = 0,
                .layerCount = 1
            }
        };
        vkCreateImageView(device, &view_info, NULL, &swap_views[i]);

        VkFramebufferCreateInfo framebuffer_info =
        {
            .sType = VK_STRUCTURE_TYPE_FRAMEBUFFER_CREATE_INFO,
            .renderPass = render_pass,
            .attachmentCount = 1,
            .pAttachments = &swap_views[i],
            .width = 1280,
            .height = 720,
            .layers = 1
        };
        vkCreateFramebuffer(device, &framebuffer_info, NULL, &framebuffers[i]);
    }

    unsigned frames_in_flight = (n_swap_images<3 ? 3 : n_swap_images);

    VkCommandPool *command_pools = malloc(frames_in_flight*sizeof(VkCommandPool));
    VkCommandBuffer *command_buffers = malloc(frames_in_flight*sizeof(VkCommandBuffer));
    for(unsigned i=0; i<frames_in_flight; ++i)
    {
        VkCommandPoolCreateInfo pool_info =
        {
            .sType = VK_STRUCTURE_TYPE_COMMAND_POOL_CREATE_INFO,
            .flags = VK_COMMAND_POOL_CREATE_TRANSIENT_BIT,
            .queueFamilyIndex = gfx_queue_index
        };
        vkCreateCommandPool(device, &pool_info, NULL, &command_pools[i]);

        VkCommandBufferAllocateInfo alloc_info =
        {
            .sType = VK_STRUCTURE_TYPE_COMMAND_BUFFER_ALLOCATE_INFO,
            .commandPool = command_pools[i],
            .level = VK_COMMAND_BUFFER_LEVEL_PRIMARY,
            .commandBufferCount = 1
        };
        vkAllocateCommandBuffers(device, &alloc_info, &command_buffers[i]);
    }

    VkSemaphore *semaphores = malloc((frames_in_flight*2)*sizeof(VkSemaphore));
    for(unsigned i=0; i<frames_in_flight*2; ++i)
    {
        VkSemaphoreCreateInfo semaphore_info =
        {
            .sType = VK_STRUCTURE_TYPE_SEMAPHORE_CREATE_INFO
        };
        vkCreateSemaphore(device, &semaphore_info, NULL, &semaphores[i]);
    }

    VkFence *fences = malloc(frames_in_flight*sizeof(VkFence));
    for(unsigned i=0; i<frames_in_flight; ++i)
    {
        VkFenceCreateInfo fence_info =
        {
            .sType = VK_STRUCTURE_TYPE_FENCE_CREATE_INFO,
            .flags = VK_FENCE_CREATE_SIGNALED_BIT
        };
        vkCreateFence(device, &fence_info, NULL, &fences[i]);
    }

    VkShaderModuleCreateInfo module_info =
    {
        .sType = VK_STRUCTURE_TYPE_SHADER_MODULE_CREATE_INFO,
        .codeSize = sizeof(shader_binary),
        .pCode = shader_binary
    };

    VkShaderModule module;
    vkCreateShaderModule(device, &module_info, NULL, &module);

    VkPipelineLayoutCreateInfo layout_info =
    {
        .sType = VK_STRUCTURE_TYPE_PIPELINE_LAYOUT_CREATE_INFO
    };

    VkPipelineLayout layout;
    vkCreatePipelineLayout(device, &layout_info, NULL, &layout);

    VkPipelineShaderStageCreateInfo stage_info[2] =
    {
        {
            .sType = VK_STRUCTURE_TYPE_PIPELINE_SHADER_STAGE_CREATE_INFO,
            .stage = VK_SHADER_STAGE_VERTEX_BIT,
            .module = module,
            .pName = "main"
        },
        {
            .sType = VK_STRUCTURE_TYPE_PIPELINE_SHADER_STAGE_CREATE_INFO,
            .stage = VK_SHADER_STAGE_FRAGMENT_BIT,
            .module = module,
            .pName = "main"
        }
    };

    VkVertexInputBindingDescription vertex_input_binding =
    {
        .binding = 0,
        .stride = 5*sizeof(float),
        .inputRate = VK_VERTEX_INPUT_RATE_VERTEX
    };

    VkVertexInputAttributeDescription vertex_attribs[2] =
    {
        {
            .location = 0,
            .binding = 0,
            .format = VK_FORMAT_R32G32_SFLOAT,
            .offset = 0
        },
        {
            .location = 1,
            .binding = 0,
            .format = VK_FORMAT_R32G32B32_SFLOAT,
            .offset = 2*sizeof(float)
        }
    };

    VkPipelineVertexInputStateCreateInfo vertex_input_state =
    {
        .sType = VK_STRUCTURE_TYPE_PIPELINE_VERTEX_INPUT_STATE_CREATE_INFO,
        .vertexBindingDescriptionCount = 1,
        .pVertexBindingDescriptions = &vertex_input_binding,
        .vertexAttributeDescriptionCount = 2,
        .pVertexAttributeDescriptions = vertex_attribs
    };

    VkPipelineInputAssemblyStateCreateInfo input_assembly_info =
    {
        .sType = VK_STRUCTURE_TYPE_PIPELINE_INPUT_ASSEMBLY_STATE_CREATE_INFO,
        .topology = VK_PRIMITIVE_TOPOLOGY_TRIANGLE_LIST,
        .primitiveRestartEnable = VK_FALSE
    };

    VkPipelineViewportStateCreateInfo viewport_info =
    {
        .sType = VK_STRUCTURE_TYPE_PIPELINE_VIEWPORT_STATE_CREATE_INFO,
        .viewportCount = 1,
        .pViewports = NULL,
        .scissorCount = 1,
        .pScissors = NULL
    };

    VkPipelineRasterizationStateCreateInfo raster_info =
    {
        .sType = VK_STRUCTURE_TYPE_PIPELINE_RASTERIZATION_STATE_CREATE_INFO,
        .depthClampEnable = VK_FALSE,
        .rasterizerDiscardEnable = VK_FALSE,
        .polygonMode = VK_POLYGON_MODE_FILL,
        .cullMode = VK_CULL_MODE_NONE,
        .frontFace = VK_FRONT_FACE_COUNTER_CLOCKWISE,
        .depthBiasEnable = VK_FALSE,
        .depthBiasConstantFactor = 0.0f,
        .depthBiasClamp = 0.0f,
        .depthBiasSlopeFactor = 0.0f,
        .lineWidth = 1.0f
    };

    VkPipelineMultisampleStateCreateInfo multisample_info =
    {
        .sType = VK_STRUCTURE_TYPE_PIPELINE_MULTISAMPLE_STATE_CREATE_INFO,
        .rasterizationSamples = VK_SAMPLE_COUNT_1_BIT,
        .sampleShadingEnable = VK_FALSE,
        .minSampleShading = 1.0f,
        .pSampleMask = 0,
        .alphaToCoverageEnable = VK_FALSE,
        .alphaToOneEnable = VK_FALSE
    };

    VkPipelineDepthStencilStateCreateInfo depth_stencil_info =
    {
        .sType = VK_STRUCTURE_TYPE_PIPELINE_DEPTH_STENCIL_STATE_CREATE_INFO,
        .depthTestEnable = VK_FALSE,
        .depthWriteEnable = VK_FALSE,
        .depthCompareOp = VK_COMPARE_OP_LESS_OR_EQUAL,
        .depthBoundsTestEnable = VK_FALSE,
        .stencilTestEnable = VK_FALSE,
        .front =
        {
            .failOp = VK_STENCIL_OP_KEEP,
            .passOp = VK_STENCIL_OP_KEEP,
            .depthFailOp = VK_STENCIL_OP_KEEP,
            .compareOp = VK_COMPARE_OP_EQUAL,
            .compareMask = 0xFFFFFFFFu,
            .writeMask = 0xFFFFFFFFu,
            .reference = 0
        }
    };
    depth_stencil_info.front = depth_stencil_info.back;

    VkPipelineColorBlendAttachmentState blend_attachment =
    {
        .blendEnable = VK_FALSE,
        .srcColorBlendFactor = VK_BLEND_FACTOR_ONE,
        .dstColorBlendFactor = VK_BLEND_FACTOR_ZERO,
        .colorBlendOp = VK_BLEND_OP_ADD,
        .srcAlphaBlendFactor = blend_attachment.srcColorBlendFactor,
        .dstAlphaBlendFactor = blend_attachment.dstColorBlendFactor,
        .alphaBlendOp = blend_attachment.colorBlendOp,
        .colorWriteMask = VK_COLOR_COMPONENT_R_BIT |
            VK_COLOR_COMPONENT_G_BIT | VK_COLOR_COMPONENT_B_BIT |
            VK_COLOR_COMPONENT_A_BIT
    };

    VkPipelineColorBlendStateCreateInfo blend_info =
    {
        .sType = VK_STRUCTURE_TYPE_PIPELINE_COLOR_BLEND_STATE_CREATE_INFO,
        .attachmentCount = 1,
        .pAttachments = &blend_attachment
    };

    VkDynamicState dynamic_states[2] = { VK_DYNAMIC_STATE_VIEWPORT, VK_DYNAMIC_STATE_SCISSOR };

    VkPipelineDynamicStateCreateInfo dynamic_info =
    {
        .sType = VK_STRUCTURE_TYPE_PIPELINE_DYNAMIC_STATE_CREATE_INFO,
        .dynamicStateCount = 2,
        .pDynamicStates = dynamic_states
    };

    VkGraphicsPipelineCreateInfo pipeline_info =
    {
        .sType = VK_STRUCTURE_TYPE_GRAPHICS_PIPELINE_CREATE_INFO,
        .stageCount = 2,
        .pStages = stage_info,
        .pVertexInputState = &vertex_input_state,
        .pInputAssemblyState = &input_assembly_info,
        .pViewportState = &viewport_info,
        .pRasterizationState = &raster_info,
        .pMultisampleState = &multisample_info,
        .pDepthStencilState = &depth_stencil_info,
        .pColorBlendState = &blend_info,
        .pDynamicState = &dynamic_info,
        .layout = layout,
        .renderPass = render_pass,
        .subpass = 0
    };

    VkPipeline pipeline;
    vkCreateGraphicsPipelines(device, VK_NULL_HANDLE, 1, &pipeline_info, NULL, &pipeline);

    VkBufferCreateInfo buffer_info =
    {
        .sType = VK_STRUCTURE_TYPE_BUFFER_CREATE_INFO,
        .size = sizeof(vertex_data),
        .usage = VK_BUFFER_USAGE_VERTEX_BUFFER_BIT,
        .sharingMode = VK_SHARING_MODE_EXCLUSIVE
    };

    VkBuffer vertex_buffer;
    vkCreateBuffer(device, &buffer_info, NULL, &vertex_buffer);

    VkPhysicalDeviceMemoryProperties mem_props;
    vkGetPhysicalDeviceMemoryProperties(physical_device, &mem_props);

    VkMemoryRequirements requirements;
    vkGetBufferMemoryRequirements(device, vertex_buffer, &requirements);

    VkMemoryAllocateInfo alloc_info =
    {
        .sType = VK_STRUCTURE_TYPE_MEMORY_ALLOCATE_INFO,
        .allocationSize = requirements.size
    };

    const VkMemoryPropertyFlags mem_flags = VK_MEMORY_PROPERTY_DEVICE_LOCAL_BIT |
        VK_MEMORY_PROPERTY_HOST_VISIBLE_BIT | VK_MEMORY_PROPERTY_HOST_COHERENT_BIT;
    for(unsigned i=0; i<mem_props.memoryTypeCount; ++i)
        if((mem_props.memoryTypes[i].propertyFlags&mem_flags)==mem_flags)
        {
            alloc_info.memoryTypeIndex = i;
            break;
        }

    VkDeviceMemory memory;
    vkAllocateMemory(device, &alloc_info, NULL, &memory);

    vkBindBufferMemory(device, vertex_buffer, memory, 0);

    void *mapped_address;
    vkMapMemory(device, memory, 0, VK_WHOLE_SIZE, 0, &mapped_address);
    memcpy(mapped_address, vertex_data, sizeof(vertex_data));

    XMapWindow(display, window);

    int first_frame = 1;
    unsigned frame_index = 0;
    while(1)
    {
        int pending = XPending(display);
        while(pending--)
        {
            XEvent event;
            XNextEvent(display, &event);
        }

        uint32_t image_index;
        vkAcquireNextImageKHR(device, swapchain, UINT64_MAX, semaphores[frame_index*2], 0, &image_index);

        vkWaitForFences(device, 1, &fences[frame_index], VK_TRUE, UINT64_MAX);
        vkResetCommandBuffer(command_buffers[frame_index], 0);
        vkResetFences(device, 1, &fences[frame_index]);

        VkCommandBuffer cmd_buf = command_buffers[frame_index];

        VkCommandBufferBeginInfo buffer_begin_info =
        {
            .sType = VK_STRUCTURE_TYPE_COMMAND_BUFFER_BEGIN_INFO,
            .flags = VK_COMMAND_BUFFER_USAGE_ONE_TIME_SUBMIT_BIT
        };

        vkBeginCommandBuffer(cmd_buf, &buffer_begin_info);

        if(first_frame)
        {
            VkBufferMemoryBarrier barrier =
            {
                .sType = VK_STRUCTURE_TYPE_BUFFER_MEMORY_BARRIER,
                .srcAccessMask = VK_ACCESS_HOST_WRITE_BIT,
                .dstAccessMask = VK_ACCESS_MEMORY_READ_BIT,
                .srcQueueFamilyIndex = VK_QUEUE_FAMILY_IGNORED,
                .dstQueueFamilyIndex = VK_QUEUE_FAMILY_IGNORED,
                .buffer = vertex_buffer,
                .offset = 0,
                .size = VK_WHOLE_SIZE
            };
            vkCmdPipelineBarrier(cmd_buf, VK_PIPELINE_STAGE_HOST_BIT, VK_PIPELINE_STAGE_VERTEX_INPUT_BIT, 0, 0, 0,
                1, &barrier, 0, NULL);

            first_frame = 0;
        }

        VkClearValue clear_value = { .color.float32 = { 0.4f, 0.4f, 0.6f, 1.0f } };

        VkRenderPassBeginInfo pass_begin_info =
        {
            .sType = VK_STRUCTURE_TYPE_RENDER_PASS_BEGIN_INFO,
            .renderPass = render_pass,
            .framebuffer = framebuffers[image_index],
            .clearValueCount = 1,
            .pClearValues = &clear_value,
            .renderArea =
            {
                {
                    .offset.x = 0,
                    .offset.y = 0
                },
                .extent.width = 1280,
                .extent.height = 720,
            }
        };
        vkCmdBeginRenderPass(cmd_buf, &pass_begin_info, VK_SUBPASS_CONTENTS_INLINE);
        vkCmdBindPipeline(cmd_buf, VK_PIPELINE_BIND_POINT_GRAPHICS, pipeline);

        VkDeviceSize vbuf_offset = 0;
        VkCmdBindVertexBuffers(cmd_buf, 0, 1, &vertex_buffer, &vbuf_offset);

        VkViewport viewport =
        {
            .width = 1280,
            .height = 720,
            .minDepth = 0.0f,
            .maxDepth = 1.0f
        };
        VkCmdSetViewport(cmd_buf, 0, 1, &viewport);

        VkRect2D scissor =
        {
            .extent.width = 1280,
            .extent.height = 720
        };
        vkCmdSetScissor(cmd_buf, 0, 1, &scissor);

        vkCmdDraw(cmd_buf, 3, 1, 0, 0);
        vkCmdEndRenderPass(cmd_buf);

        vkEndCommandBuffer(cmd_buf);

        VkPipelineStageFlags wait_stages = VK_PIPELINE_STAGE_ALL_GRAPHICS_BIT;

        VkSubmitInfo submit_info =
        {
            .sType = VK_STRUCTURE_TYPE_SUBMIT_INFO,
            .waitSemaphoreCount = 1,
            .pWaitSemaphores = &semaphores[frame_index*2],
            .pWaitDstStageMask = &wait_stages,
            .commandBufferCount = 1,
            .pCommandBuffers = &cmd_buf,
            .signalSemaphoreCount = 1,
            .pSignalSemaphores = &semaphores[frame_index*2+1]
        };
        vkQueueSubmit(graphics_queue, 1, &submit_info, fences[frame_index]);

        VkPresentInfoKHR present_info =
        {
            .sType = VK_STRUCTURE_TYPE_PRESENT_INFO_KHR,
            .waitSemaphoreCount = 1,
            .pWaitSemaphores = &semaphores[frame_index*2+1],
            .swapchainCount = 1,
            .pSwapchains = &swapchain,
            .pImageIndices = &image_index
        };
        vkQueuePresentKHR(graphics_queue, &present_info);

        frame_index = (frame_index+1)%frames_in_flight;
    }

    return 0;
}
```

Kolmion piirtävä ohjelma toteutettuna vanhalla OpenGL:llä (vasemmalla ylhäällä), modernilla OpenGL:llä (vasemmalla alhaalla) ja Vulkanilla. Visualisoinnissa käytetyt esimerkkikoodit löytyvät Skrollin verkkojatkoilta: skrolli.fi/numerot.

topuskureihin ja koko puskuri annetaan näytönohjaimelle kerralla[1]. Tämä vähentää säikeiden välisen synkronoinnin tarvetta ja parantaa suorituskykyä, koska puskurit voivat olla paikallisia niitä käsittelevälle säikeelle ja synkronointia tarvitaan vasta, kun puskuri on valmis lähetettäväksi.

Toinen suorituskykyongelmia aiheuttava piirre OpenGL:ssä on tilan rakentuminen. Eri tyyppiset resurssit sidotaan kukin erikseen osaksi kontekstin tilaa, ja niiden parametrit asetetaan yksi kerrallaan. Tästä voi muodostua monimutkaisiakin riippuvuusketjuja. Esimerkiksi shaderien kirjoittamat väriarvot saattavat mennä ruutupuskuriin (*framebuffer object*), johon puolestaan on liitetty tekstuuri, jonka tekselien tyyppi vaikuttaa väriarvojen käsittelyyn. Koska tila voi muuttua koska tahansa, ajureissa täytyy olla paljon logiikkaa oikean toiminnan varmistamiseksi.

Vulkanissa sitä vastoin suurin osa parametreistä asetetaan resurssia luotaessa, eikä resurssin rakennetta voi tämän jälkeen muuttaa. Tämä mahdollistaa monien tarkistusten tekemisen jo resurssien luonnin yhteydessä. Jos resurssi kelpaa käyttötarkoitukseensa kerran, se kelpaa myös jatkossa.

Vaatimuksia ajurien tekemille virhetarkastuksille on myös kevennetty. Jos vaikkapa ruutupuskuriin liitetyn tekstuurin värikomponenttien tyyppi ei vastaa odotuksia, shaderi voi vapaasti kirjoittaa muistiin vääränlaisia arvoja. Seurauksena on sotkua, kun tekstuuria myöhemmin käytetään piirtämiseen. Sovelluskehittäjän työn helpottamiseksi Vulkanin ydinrajapinnan päälle voi pinota virheentarkistustyökaluja – ja jättää ne taas pois kun koodi on todettu toimivaksi.

Kääntöpuolena näillä muutoksilla on sovelluskoodin monimutkaistuminen. Sanonnan mukaan tietokonegrafiikan ”hei maailma” -ohjelman toteuttaminen eli kolmion piirtäminen ruudulle vie vanhalla OpenGL:llä 10 riviä koodia, modernilla OpenGL:llä 100 riviä ja Vulkanilla 1000 riviä. Oman kokemukseni mukaan tämä pitää summittain paikkansa. Valtaosa eroista liittyy kuitenki alustuskoodiin, joka tarvitsee kirjoittaa vain kerran. Triviaalia suuremmassa Vulkan-sovelluksessa ei siis ole kymmenkertaista määrää koodia vastaavaan OpenGL-sovellukseen verrattuna.

Yhteistäkin Vulkan- ja OpenGL-rajapinnoilla on: Molempia ylläpitää voittoa tavoittelematon Khronos-ryhmä (khronos.org), ja molemmat ovat käytettävissä maksutta avoimilla lisensseillä. Khronos kutsuikin Vulkanin synnyttänyttä prosessia ”seuraavan sukupolven OpenGL-hankkeeksi” ja ”OpenGL nextiksi”, joskin näistä nimityksistä on sittemmin luovuttu.

Hyvin suunniteltu on puoliksi tehty

Vulkan-rajapinta on kiinnostanut minua sen julkaisusta lähtien, sillä olen kehittänyt omaa OpenGL-pohjaista grafiikkamoottoriani vuodesta 2006 alkaen (lisää tästä oheisessa laatikossa). Vuonna 2017 yritin kirjoittaa pientä Vulkan-testiohjelmaa puhtaalta pöydältä, mutta kyllästyin alustuskoodin paljouteen ja homma jäi kesken. Alkuvuodesta 2021 sain kuitenkin idean pelistä, jonka haluaisin toteuttaa keinotodellisuutta käyttäen, ja keinotodellisuus jos mikä hyötyy maksimaalisesta suorituskyvystä. Ehkäpä siis aika olisi kypsä uudelle yritykselle.

Oma grafiikkamoottorini oli rakennettu hyvin vahvasti OpenGL:n ympärille, joten ensi ajattelemalta sen muuntaminen käyttämään Vulkania vaikutti valtavalta urakalta. Mutta pyöriteltyäni ajatusta päässäni muutaman saunaillan verran se alkoi kuin itsestään pilkkoutua hallittaviksi osasiksi ja tie eteenpäin aukeni.

Koska OpenGL ja Vulkan ovat molemmat tarkoitettuja reaaliaikaisen tietokonegrafiikan tuottamiseen, niistä löytyy paljon yhtäläisyyksiä. Molemmat käyttävät esimerkiksi puskureita, shadereita ja tekstuureita. Semanttisia eroja tietysti on, mutta mikään grafiikkamoottorissa käyttämäni toiminnallisuus ei ole sellainen, mitä ei Vulkanista löytyisi (reilua vuotta aiemmin tehdystä vanhojen ominaisuuksien karsinnasta oli nyt hyötyä, kun minun ei tarvitsisi murehtia esimerkiksi emulaation toteuttamista).

OpenGL-tuotokseni laajentamisessa tukemaan Vulkania olisi kolme päävaihetta: Ensin muokkaisin julkisia rajapintoja poistaakseni sellaiset toiminnot, joita Vulkanissa ei ole. Sitten erottaisin OpenGL-kutsut yleisestä toiminnallisuudesta ja lopuksi toteuttaisin vaihtoehtoisen, Vulkania käyttävän piirtokoodin. Ohjelman modulaarisesta luokkarakenteesta olisi taas kerran hyötyä, koska voisin käydä läpi yhden luokan kerrallaan ja tehdä tarvittavat muutokset ilman, että koko rakennelma sortuu.

Koodin uudistumisen ei tarvitsisi jäädä Vulkaniin. Potentiaalisesti voisin tehdä rinnalle myös kaksi erillistä OpenGL-toteutusta, joista toinen olisi yhteensopiva vanhemman OpenGL 3.3:n kanssa ja toinen käyttäisi rajapinnan uusimpia ominaisuuksia. Se mahdollistaisi erityisesti OpenGL:n `ARB_direct_state_access`-laajennoksen (*extension*) paremman hyödyntämisen – laajennos tarjoaa resurssien parametreja tai sisältöä muuttavista funktioista versiot, jotka operoivat suoraan nimillä tarvitsematta sitoa resurssia ensin käytettäväksi. Koodista tulee virtaviivaisempaa, mutta koska laajennos lisättiin osaksi OpenGL:n ydinrajapintaa vasta versiossa 4.5, hyöty vesittyy, jos myös vanhempia versioita pitää tukea.

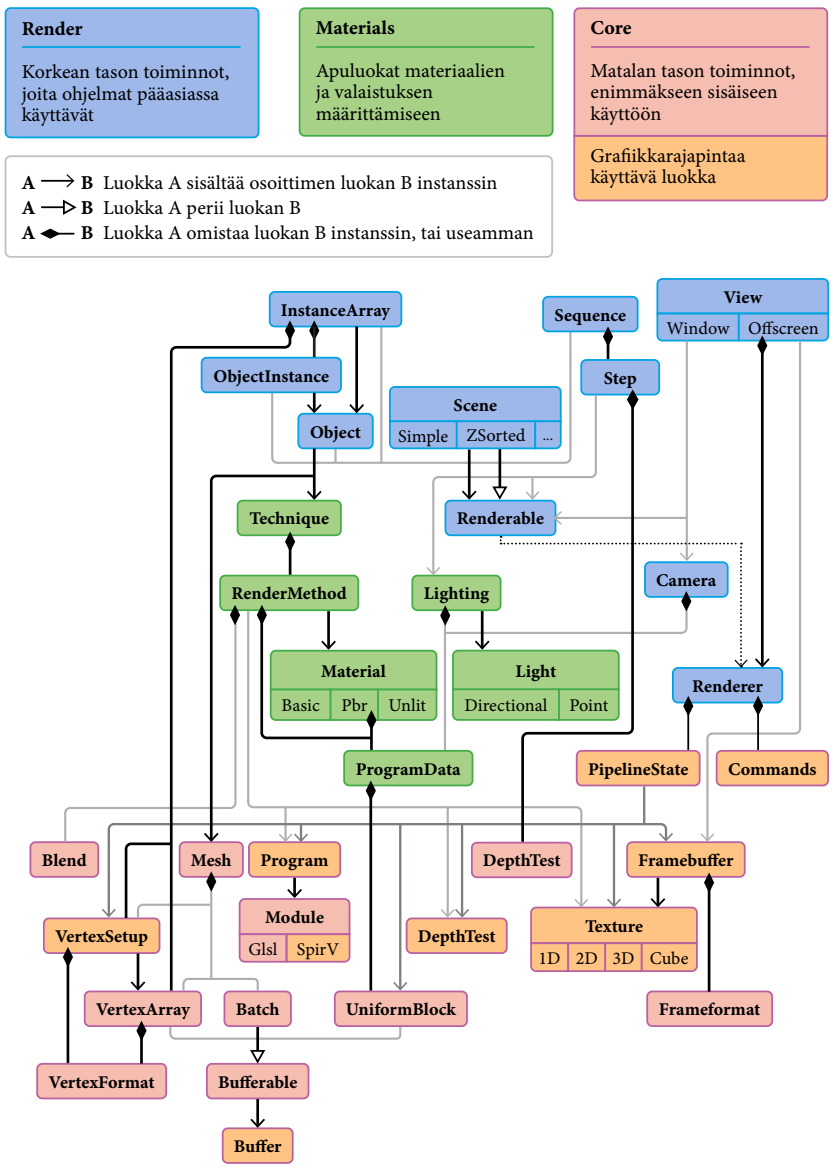
Kohti Vulkania

Koodin kimppuun siis. Ensimmäisenä Vulkan-yhteensopivuuden lisäämisessä päätin ottaa käsittelyyn puskurit (*buffer*), joita käytetään erityisesti kolmioverkkojen kulmapisteiden säilyttämiseen. Vaikka tavoite oli selvä, asiassa riitti yksityiskohtia pohdittavaksi. Eräs keskeisimpiä luokkia koodissa on `Bufferable`, joka toimii joidenkin muiden luokkien kantaluokkana ja avustaa datan siirtämisessä puskureihin. Tähän asti puskuri oli annettu ulkopuolelta ja oliot olivat huolehtineet puskurin koon muuttamisesta. Vulkanissa koon muutos ei kuitenkaan olisi mahdollista.

Aluksi kokeilin muuttaa rajapintaa siten, että `Bufferable` ei muuttaisi puskurin kokoa vaan heittäisi poikkeuksen, jos sille annettu puskuri on liian pieni. Tämä johti kuitenkin ylettömän monimutkaiseen koodiin. Suurin syy oli mahdollisuus pakata samaan puskuriin useita olioita siten, että ne muodostavat keskenään linkitetyn listan. En keksinyt eleganttia tapaa tarkistaa etukäteen, onko puskuri riittävän suuri uuden olion lisäämiselle. Talletin keskeneräisen koodin varmuuden vuoksi versionhallinnan haaraan ja yritin uudestaan.

Toinen lähestymistapa oli piilottaa puskurit kokonaan `Bufferable`-luokan sisään. Suunnitteluhermoani kutkutti ajatus siitä, ettei muiden luokkien tarvitsisi enää huolehtia puskureiden

[1] *Todellisuudessa OpenGL-ajurit tekevät samanlaista puskurointia kulissien takana, mutta tämä tapahtuu väärällä puolella rajapintaa auttaakseen säikeistyksessä.*



Taustalla OpenGL

Vaikka OpenGL ei varsinaisesti ole Vulkanin taustalla, se on monen Vulkanin käyttäjän taustalla.

Aloitin nykyisen oman 3D-grafiikkamoottorini tekemisen vuonna 2006. Se on järjestyksessä kolmas: ensimmäisen ja toisen hylkäsin, koska ohjelmointitaitojeni kehittyessä havaitsin niissä olevan suunnitteluvirheitä, joiden korjaaminen olisi ollut hyvin työlästä. Mutta kolmas kerta toden sanoo, ja tästä versiosta onnistuin suunnittelemaan niin joustavan ja laajennettavan, ettei täydelliselle uudelleenkirjoittamiselle ole ollut tarvetta.

Viisitoista vuotta sitten tietokonegrafiikan pelikenttä näytti hyvin erilaiselta. Avoimia vaihtoehtoja oli vähän ja ne olivat vielä raakileita. Suurten kaupallisten grafiikkamoottorien ilmaisversiot olivat vielä lähes vuosikymmenen päässä. Myös näytönohjaimet olivat yksinkertaisempia – vapaasti ohjelmoitavat shaderit oli esitelty vasta pari vuotta aiemmin. Motivaatio oman moottorin aloittamiseen oli siis suurempi ja kynnys matalampi kuin nyt.

OpenGL oli luonnollinen valinta grafiikkamoottorin toteutusrajapinnaksi sen laajan alustatuen takia. Ainoa varteenotettava vaihtoehto oli Direct3D, mutta Microsoftin alustoihin rajoittuneena se ei houkutellut. Tuolloin en odottanut tarvitsevani tukea useille eri grafiikkarajapinnoille, joten pidin koodin yksinkertaisena ja toteutin toiminnallisuuden suoraan julkisiin luokkiin ilman ylimääräisiä abstraktioita.

Aluksi grafiikkamoottorini ei ollut muuta kuin joukko enimmäkseen irrallisia luokkia OpenGL:n resurssien käsittelyn helpottamiseksi. Niiden muodostamalle perustalle aloin rakentaa aina vain kehittyneempiä ominaisuuksia. Arkkitehtuuri on hierarkkinen: kukin luokka joko toteuttaa jonkin matalan tason ominaisuuden tai yhdistelee alemman tason luokkia. Viittauksia ylemmän tason luokkiin ei ole. Tämän modulaarisen rakenteen avulla olen saanut pidettyä moottorin joustavana ja muokattavana.

Kun aloittelin moottorin toteutusta, OpenGL-piireissä pöhistiin tulevasta Longs Peak -koodinimellä tunnetusta versiosta, jonka piti uudistaa rajapintaa rankalla kädellä. Resurssien luonnista tulisi atomista, eikä niiden rakennetta voisi muuttaa luomisen jälkeen. Lopulta vuoden myöhässä julkaistu OpenGL 3.0 oli paljon odotettua laimeampi uudistus, mutta se sentään avasi oven vanhentuneiden toiminnallisuuksien tuen poistamiselle. Longs Peakiin luvatut massiiviset uudistukset saatiin vasta Vulkan-rajapinnan julkaisussa reilu seitsemän vuotta myöhemmin.

Longs Peak ehti vaikuttaa omaan grafiikkamoottoriini siten, että tekstuurien kokoa ja tekselien tyyppiä ei voi muuttaa, kun se on kerran määritelty. Jälkikäteen ajateltuna olisin voinut noudattaa samaa ajatusta myös ainakin puskureissa.

Olen aiemminkin muokannut moottoria modernimpaan suuntaan. Ensimmäinen laajamittainen muutos oli OpenGL ES -tuen lisääminen vuosina 2012–2014. Vuonna 2019 poistin lopulta kaiken tuen vanhalle OpenGL 1.x:n aikaiselle kiinteästi määritellylle toiminnallisuudelle. Vulkan-tuen lisääminen on kuitenkin ylivoimaisesti suurin yksittäinen muutosprojekti tähän asti.

Pienempiä parannuksia

Vulkan-projektin aikana tein myös lukuisia pienempiä parannuksia grafiikkamoottorini ominaisuuksiin. Listaan tässä lyhyesti joitakin niistä.

Lisäsin efektin, joka piirtää taustalle taivaan käyttäen approksimaatiota valon siroamisesta ilmakehässä. Auringon asentoon dynaamisesti reagoiva taivas elävöittää näkymää.

Käytän sisällön luomiseen Blenderiä (blender.org), jolle olen tehnyt oman lisäosan mallien viemiseksi omaan tiedostomuotooni. Parantelin tätä lisäosaa ja lisäsin siihen mahdollisuuden kokonaisen projektin viemiseksi kerralla. Työskentely helpottuu, kun yksi nappi hoitaa kaiken.

Kulmapisteiden koordinaatit ja muut arvot olivat tähän mennessä voineet olla ainoastaan 32-bittisiä liukulukuja, poikkeuksena RGBA-väriarvot. Laajensin kulmapisteiden rakenteen kuvausta tukemaan mielivaltaisia tyyppejä. Monesti esimerkiksi tekstuurikoordinaateille tai normaaleille riittää 16- tai jopa 8-bittinen tarkkuus. Kompaktimpi esitys säästää muistia ja parantaa suorituskykyä.

Toteutin pistemäisille valonlähteille varjot nelitahoisilla varjokartoilla (*tetrahedral shadow map*). Samalla lisäsin tuen varjokartastolle (*shadow atlas*), jossa useita varjokarttoja pakataan samaan tekstuuriin, kukin omalle alueelleen.

varaamisesta. Törmäsin kuitenkin tässäkin ongelmiin, sillä kulmapisteiden rakenteesta huolehtiva `VertexSetup` (vastaa OpenGL:n vertex array objectia) tarvitsee puskuria, johon kulmapisteet on tallennettu. Jos puskuri voidaan luoda uudelleen koska tahansa, sitä olisi vaikeaa pitää ajan tasalla tehokkaasti. Hankaluuksia olisi luvassa myös, jos haluaisin käyttää laskentashadereita (*compute shader*) kulmapisteiden päivittämiseen, koska tällöin puskurin luonnissa pitäisi käyttää hieman tavallisesta poikkeavia parametreja. Tämäkin ratkaisu lensi romukoppaan.

Lopulta päädyin toteuttamaan ensimmäisestä ratkaisusta version, jossa kaikkia epäkelpoja tiloja ei yritetä estää, vaan poikkeus heitetään vasta, jos kirjoitettava data ei mahdu puskuriin. Näin puskurin omistava olio voi kaikessa rauhassa koostaa puskuriin päätyvät oliot ja vasta lopuksi kysyä niiltä, kuinka suuri puskurin pitäisi olla.

Myös pari pienempää muutosta oli tarpeen. OpenGL:n puskureille voi antaa vihjeen puskurin käyttötavasta, mutta juuri mikään ajuri ei tehnyt sillä mitään ja Vulkanissa koko vihjettä ei enää ole. Puskurin sisältöä sovelluksen muistiavaruuteen liitettäessä (*memory map*) käytettävä siirtosuuntaa ilmaiseva parametri jäi niin ikään tarpeettomaksi.

Puskurin koon voi muuten OpenGL:ssäkin lukita käyttämällä versiossa 4.4 lisättyä `glBufferStorage`-kutsua. Vanhempiin versioihin se on saatavilla `ARB_buffer_storage`-laajennoksen kautta.

Tekstuurit ja näytteistimet

Seuraavaksi työpöydälle päätyivät tekstuurit. Vulkan vaatii myös niiden rakenteen olevan muuttumaton, mutta tämän olinkin toteuttanut jo aiemmin – OpenGL 4.2:ssa esiteltiin `glTexStorage*`-kutsut, joilla rakenteen voi lukita myös rajapinnan tasolla, ja `ARB_texture_storage`-laajennos tarjoaa vastaavan toiminnallisuuden vanhempiin OpenGL-versioihin.

Toinen merkittävä ero Vulkanissa liittyy näytteistysasetuksiin, joita ovat muun muassa väriarvojen suodatus ja tekstuurikoordinaattien tulkinta perusvälin [0, 1] ulkopuolella. OpenGL:ssä nämä ovat osa tekstuurin tilaa, mutta Vulkan vaatii erillisten näytteistimien käyttöä. Koska halusin oman moottorini matalan tason luokkien likimain vastaavan grafiikka-alustan toimintoja, siirsin näytteistimiin liittyvän toiminnallisuuden `Texture`-luokasta uuteen `Sampler`-luokkaan.

Näytteistimien lisäys vaikutti epätriviaalilla tavalla myös joihinkin muihin luokkiin. `PostProcessor`- ja `Effect`-luokista johdetut efektit käyttävät tekstuureita, joten ne tarvitsivat nyt myös näytteistimet. Koska monet efektit käyttävät samanlaisia näytteistimiä, aioin aluksi käyttää jaettuja osoittimia. Muutosta tehdessäni tajusin vastaavia sisäänrakennettuja resursseja olevan niin paljon, että olisi järkevintä valjastaa moottorin käyttämä resurssienlatausmekanismi tähän tarkoitukseen.

Seurauksena edellisestä efekti-instanssien luominen vaati viittauksen resurssisäiliöön. Suoraviivaisin tapa oli antaa sellainen argumenttina luo-

kan rakentajalle. Se oli kuitenkin epäkäytännöllistä, joissain tapauksissa jopa ongelmallista, eikä istunut kovin hyvin moottorin yleiseen suunnitteluun. Myöhemmin lisäsin `Resources`-luokkaan staattisen osoittimen, jonka kautta efektit pääsevät käsiksi ensimmäiseen sovelluksen luomaan resurssisäiliöön.

`Material`-luokan kohdalla mietin, pitäisikö jokaiselle materiaalin tekstuurille voida määrittää näytteistin erikseen vai riittäisikö yksi näytteistin koko materiaalille. Päädyin lopulta käyttämään materiaalin laajuista näytteistintä, koska en keksinyt käyttötapausta, jossa tekstuurikohtaisista näytteistimistä olisi etua. Materiaalit ladataan jo ennestään resurssitiedostoista, joten myös näytteistimen lataaminen sitä kautta oli luonnollinen ratkaisu.

Tekstuurien tekselien tyypin on OpenGL:ssä perinteisesti voinut määritellä pelkästään värikomponenttien tasolla ja antaa implementaation valita bittisyvyys. Vulkanissa tämä ei käy, vaan tyyppi on määriteltävä täsmällisesti. Pienellä enumeraatioarvojen uudelleenjärjestelyllä sain oman rajapintani vastaamaan tätä vaatimusta.

OpenGL:ään näytteistimet lisättiin versiossa 3.3, ja vanhempiin versioihin sen saa `ARB_sampler_objects`-laajennoksella – tosin käytännössä kaikki viimeisen 10 vuoden aikana julkaistut näytönohjaimet tukevat OpenGL 3.3:a.

Shaderit

Nyt kun projektin rakenne oli palautettu mieleen ja aivojen koodauslohkot lämmitelty, oli aika käydä käsiksi yhteen tärkeimmistä osa-alueista eli shadereihin. OpenGL:ssä shaderit luodaan GLSL-kielisestä lähdekoodista, joka käännetään ajurissa näytönohjaimen binäärikoodiksi. Tämä monimutkaistaa ajureita ja saattaa tehdä suurien shaderien lataamisesta hidasta. Joidenkin valmistajien ajurit myös sallivat tiettyjä poikkeamia spesifikaatiosta, joten yhdellä näytönohjaimella testattu shaderi ei välttämättä toimi toisella.

Vulkanissa sen sijaan GLSL-koodi käännetään etukäteen erillisellä työkalulla SPIR-V-välikoodiksi, joka voidaan sitten ajon aikana muuttaa näytönohjainkohtaiseksi binääriksi. Välikoodia on yksinkertaisempaa ja nopeampaa käsitellä, koska se on määritelty hyvin täsmällisesti ja sisältää vain yksinkertaisia operaatioita monimutkaisten lausekkeiden sijaan. (Myös OpenGL:ssä on mahdollista käyttää SPIR-V-shadereita versiossa 4.6 tai `ARB_gl_spirv`-laajennoksen avulla.)

Yksi SPIR-V-binääri voi sisältää useita shaderivaiheita, ja samaa binääriä on mahdollista erikoistaa latauksen yhteydessä. Tämä poikkeaa GLSL-shadereista, joissa jokainen vaihe käännetään erikseen omasta lähdekoodistaan. Vanha `Shader`-luokka sai siis lähteä, ja tilalle tuli `Module`, josta on peritty luokat GLSL- ja SPIR-V-muotoisia moduuleja varten. `Program`-luokan rajapinta muuttui myös vastaavasti.

Erillisen käännösvaiheen takia Vulkanin SPIR-V-shaderien käyttö on hieman työläämpää kuin OpenGL:n GLSL-shaderien. Erityisesti iteratiivisessa kehityksessä on epäkäytännöllistä, jos jokaisen muutoksen jälkeen ohjelma täytyy kääntää uudestaan. Tästä syystä grafiikkamoottorini osaa tarvittaessa kääntää GLSL-lähdekoodin lennossa SPIR-V-moduuliksi.

Vulkan vaatii, että kaikki tavalliset uniform-muuttujat on sijoitettu lohkoihin (*uniform block*) ja niille annetaan arvot puskurien kautta. OpenGL:stä tuttua tapaa asettaa yksittäisen muuttujan arvo suoraan ei ole käytettävissä. Puskurin sisällön on myös pysyttävä samana, kunnes sitä käyttävät piirtokomennot on suoritettu, mikä käytännössä tarkoittaa, että animoituja arvoja sisältävistä lohkoista on oltava muistissa useita kopioita. Oma grafiikkamoottorini on tukenut uniform-lohkoja jo pitkään, mutta moninkertaistus animointia varten vaati joitain muutoksia (lisää jäljempänä).

Erikoistapauksena yhden lohkon voi merkata `push_constant`-määreellä, jolloin sen sisältö määritellään puskurin sijaan piirtokomentojen yhteydessä. Näiden arvojen käyttö shaderissa on hyvin nopeaa, mutta myös tilaa on erittäin rajallisesti – Vulkan-spesifikaatio ei lupaa enempää kuin 128 tavua, joka riittää vain 32 liukulukuarvolle. Käytännössä näytönohjaimet tarjoavat 256–4096 tavua tilaa.

Tekstuurien lukemiseen käytettävät `sampler`-tyyppiset muuttujat määritellään yhä globaalisti lohkojen ulkopuolella, mutta käytettävän tekstuuriyksikön indeksiä ei voi enää muuttaa rajapinnan kautta. Sen sijaan se on asetettava shaderin lähdekoodissa `binding`-määreellä.

Muutin oman grafiikkamoottorini toimintaa siten, että tekstuurien asettaminen käyttöön tapahtuu tekstuuriyksikön numeron sijaan shaderissa olevan muuttujan nimen perusteella. Se vaati merkittäviä muutoksia erityisesti `RenderMethod`- ja `Renderer`-luokkiin. Teksturointitilan säilyttämiseen käytetty `Texturing`-luokka jäi pois: yksittäisiä tekstuureja käsitellään nyt suoraan. Tämä tekee moottorista joustavamman tilanteissa, joissa tekstuureja tulee useammasta lähteestä, kuten monien efektien kanssa.

Yleiskäyttöisen grafiikkamoottorin on tärkeää kyetä käyttämään laajaa joukkoa erilaisia shadereita ja antamaan niille syötteenä oikeat muuttujat. Koska moottori ei voi etukäteen tietää jokaisen mahdollisen shaderin rajapintaa, sen on luettava muuttujien tiedot shaderista itsestään. OpenGL tarjoaa tähän reflektioksi kutsuttuun toimenpiteeseen joukon funktioita, mutta Vulkania käytettäessä tiedot on itse ongittava SPIR-V-binääristä. Formaatti on onneksi varsin selkeä. Tarjolla on myös valmiita kirjastoja.

Tilan hallinta

Kuten artikkelin alussa mainitsin, OpenGL:ssä on paljon piirtokomentoihin vaikuttavaa globaalia tilaa. Eri resurssit ovat pääasiassa toisistaan erillisiä, ja ainoastaan parissa tapauksessa (ruutupuskurit, verteksitaulukot) useita tilamuuttujia voi niputtaa yhteen resurssiin. Juuriensa takia oma grafiikkamoottorini noudatti samaa linjaa ja tarjosi julkisessa rajapinnassaan funktiot eri olioiden sitomiseen.

Vulkanissa mikään tila ei ole globaalia, vaan sekä käytettävien resurssien sidonnat että piirtokomennot kirjataan komentopuskureihin. Uuden komentopuskurin täyttäminen aloitetaan aina puhtaalta pöydältä. Näin esimerkiksi kolmannen osapuolen kirjastot eivät pääse sotkemaan asioita jättämällä jonkin tilan väärään asentoon. Keskeisellä paikalla tilan asettamisessa on linjastoolio (*pipeline state object*), joka pitää sisällään käytettävän shaderin, kuvauksen syötteiden ja tulosten rakenteesta sekä grafiikkaprosessorin kiinteän toiminnallisuuden konfiguraation.

Grafiikkamoottorissani on jo pitkään ollut `Renderer`-luokka, joka toimii korkean tason rajapintana tilan asettamiselle ja pyrkii vähentämään matalan tason kutsujen määrää. Ei siis ollut isokaan askel tiputtaa julkisia sidontafunktioita kokonaan pois muista luokista. Renderer toimii kuitenkin turhan korkealla abstraktiotasolla voidakseen tehokkaasti hallita linjasto-olioiden luomista. Tarvitsin uuden luokan, joka sai nimekseen `PipelineState`.

Grafiikka-alustasta riippumatta resurssien asettaminen käyttöön vaatii matalan tason tunnisteen käyttöä. En halunnut paljastaa niitä julkisessa rajapinnassa, koska se aiheuttaisi vuodon abstraktioon ja sallisi sovellusten tehdä tyhmyyksiä. Ratkaisin asian käyttämällä ystävämäärityksiä (`friend`); yleensä vältän niitä, koska pääsyä ei ole mahdollista rajoittaa mitenkään, mutta tässä tapauksessa koin sen olevan pienempi paha.

Aioin aluksi toteuttaa OpenGL-versioon sidontamekanismin, jossa uusi linjasto-olio korvaisi edellisen tilan kokonaan. Siihen tarvittavasta koodista tuli kuitenkin varsin monimutkaista, ja tajusin, ettei sille ole tarvetta kesken ruudun renderöinnin – ei ole väliä, jos pari ylimääräistä tekstuuria on käytössä sidontapisteissä, joita nykyinen shaderi ei tarvitse. Riittää, että ruudun valmistuttua sidonnat puretaan, jotta resurssien tuhoaminen

Oma GLSL-kääntäjä

Grafiikkamoottorissani on jo pitkään ollut mahdollisuus luoda perusshadereita antamalla yksinkertainen kuvaus siitä, mitä ominaisuuksia (esimerkiksi teksturointi, valaistus, normaalikartta) shaderin tulisi tukea. Vuosien varrella olen tehnyt tästä kolme eri toteutusta, joista viimeisin perustuu GLSL-koodin muokkaukseen omalla kääntäjällä. Perustapauksessa kääntäjälle annetaan syötteenä kaikki ominaisuudet sisältävä mallineshaderi ja tieto siitä, mitä ominaisuuksia halutaan käyttää. Näistä lähtökohdista kääntäjä tuottaa optimoidun shaderin, jossa on vain halutut ominaisuudet.

Ominaisuuksien valinnan lisäksi kääntäjässä on muitakin omia lisäyksiä GLSL:ään. Shaderiohjelman kaikki vaiheet voi kirjoittaa samaan tiedostoon kokonaisuuden hallinnan helpottamiseksi. Automaattinen rajapintojen generointi helpottaa arvojen välittämistä vaiheesta toiseen. Moduulien ja tuontimekanismin (*import*) avulla välttyy kirjoittamasta usein tarvittavia pätkiä moneen kertaan. Funktioita uudelleenmäärittelemällä on mahdollista hienosäätää perusshaderien toimintaa.

Alun perin tämä oma kääntäjä toimi ainoastaan GLSL-esikäsittelijänä ja jätti koodin tarkemman tulkinnan ja virheenkäsittelyn ajureiden kääntäjälle. Nyt halusin kehittää siitä itsenäisen kääntäjän, joka pystyisi kääntämään GLSL-lähdekoodin aina SPIR-V-binääriksi asti. Kaikeksi onneksi olin tehnyt pohjatyön kunnolla ja perusedellytykset kuten lähdekoodin jäsennys syntaksipuuksi olivat olemassa. Kääntäjän arkkitehtuuri perustuu vierailijaluokkiin (visitor-kaava), joten uusien operaatioiden lisäys oli myös helppoa.

Yksi suurimpia puutteita oli se, ettei kääntäjä ymmärtänyt tietotyyppejä. Se kyllä tunnisti tyyppien nimet ja tallensi ne syntaksipuuhun, mutta ei ymmärtänyt mitä ominaisuuksia eri tyypeillä on tai mitä tyyppiä lausekkeiden tulokset ovat. SPIR-V vaatii moduulia määrittelemään kaikki käyttämänsä tyypit, joten lisäsin myös omassa kääntäjässäni kaikki tyyppimäärittelyt osaksi syntaksipuuta. Kääntäjässä on sisäänrakennettuna GLSL:n vakiotyypit, joiden määrittelyt lisätään syntaksipuuhun käännöksen alussa.

Olin oikaissut pari mutkaa lausekkeiden kanssa ja jättänyt tekemättä erilliset syntaksipuun solmutyypit sekoitukselle (*swizzle*) ja valintaoperaattorille (:?). Sekoitus käyttää samaa syntaksia kuin struktuurien jäsenten käsittely, joten sitä ei edes ollut mahdollista tunnistaa ilman ymmärrystä tyypeistä. Sitä ei liioin voi tunnistaa vielä jäsennysvaiheessa, koska vasemmalla puolella olevan lausekkeen tyyppi on tunnettava ensin.

Kääntäjässä ei myöskään ollut tietoa GLSL:n sisäänrakennetuista funktioista, vaan se oletti kaikkien tuntemattomien funktioiden olevan sisäänrakennettuja. Se oli ongelmallista sekä lausekkeiden tyyppien määrittämisen että koodin oikeellisuuden tarkistamisen kannalta. Koska sisäänrakennettuja funktioita on runsaasti (tällä hetkellä kääntäjäni tuntee niitä lähes 500 ja osa puuttuu vielä) ja lähes kaikilla on useita ylikuormituksia eri tyypeille, kirjoitin Python-skriptin funktiomäärittelyjen luomista varten.

Toiminnallisten puutteiden korjaamisen lisäksi toteutin kääntäjään myös uusia optimointeja. Aiemmin funktioiden upotus (*inlining*) oli mahdollista vain, jos funktio ei ottanut parametreja ja koostui yhdestä return-lauseesta. Nyt parametrien upotus onnistuu ja funktiot saavat olla pidempiä. Lisäksi jos sijoitusoperaattorin vasemmalla puolella olevaa muuttujaa käytetään vain kerran, sijoitettava lauseke voidaan upottaa muuttujan käyttöpaikkaan. Näiden parannusten jälkeen monet shaderit optimoituvat yhdeksi funktioksi, joka sisältää muutaman pitkän lausekkeen.

SPIR-V:ssä on mahdollista määritellä erikoistamisvakioita, joiden arvoa voidaan muuttaa moduulia ladattaessa. Minulla oli ennestään

myöhemmin myös vapauttaa niiden käyttämän muistin.

Lisäsin myös piirtokomentoja varten uuden `Commands`-luokan. OpenGL:ssä sen merkitys on vähäinen, mutta Vulkanissa se tulisi hallitsemaan komentopuskureita ja kirjaamaan komennot niihin.

Irti OpenGL:stä

Yhdeksän kuukautta projektin aloittamisen jälkeen tarvittavat muutokset toiminnallisuuteen oli lopulta tehty ja pääsin etenemään suunnitelman toiseen vaiheeseen. Kaikki OpenGL:ää käyttävä koodi piti eriyttää siitä riippumattomasta koodista. Mahdollisia toteutustapoja on muutama erilainen, joista kaikilla on omat vahvuutensa.

Perinteinen olio-ohjelmointia hyödyntävä tapa on määritellä jokaiselle luokalle erillinen abstrakti rajapinta ja kutakin alustaa vastaava toteutus. Tämä on erittäin joustava toteutustapa ja mahdollistaa taustakoodin valinnan ohjelman suorituksen aikana. Kääntöpuolena kaikkien alustariippuvaisten luokkien instanssit on luotava tehdasfunktioiden kautta ja erilliset rajapinnat vaativat ylimääräistä koodia.

Toinen yleisesti käytetty tapa on pitää luokkien julkinen rajapinta ennallaan mutta piilottaa alustariippuvaiset osat osoittimen taakse (niin sanottu pimpl-kaava). Instansseja voi luoda normaaliin tapaan, mutta alusta on valittava käännöstä tehdessä. Olioiden sisältämä

hyvin samankaltainen mekanismi, jossa muuttujia saattoi shaderin sisällä määritellä uudelleen uusilla arvoilla. Se oli melko helppo muuttaa käyttämään SPIR-V:n ominaisuuksia vastaavaa syntaksia.

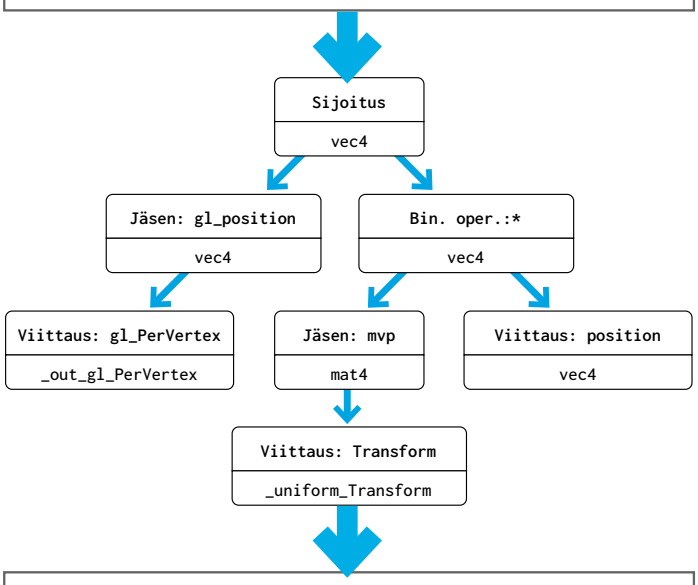
Virheentarkistusta kääntäjässä ei aiemmin ollut lainkaan. Se oli lisättävä, sillä halusin kääntäjän tuottavan SPIR-V-binäärejä, enkä siten voinut luottaa alemmalla tasolla tapahtuvaan virheentarkistukseen. Toteutus vaati hieman pohdintaa, koska halusin heittää poikkeuksen, jos kääntäjälle annettu lähdekoodi on virheellinen, mutta toisaalta halusin raportoida käyttäjälle kaikki virheet enkä vain ensimmäistä. Päädyin ratkaisuun, jossa virheet kerätään prosessoinnin aikana syntaksipuun yhteyteen ja kääntäjän päätaso heittää poikkeuksen, jos virheitä löytyi.

Kehityksen aikana nousi esiin hyvin konkreettinen huoli siitä, rikonko joitain aiemmin toimineita ominaisuuksia. Otin härkää sarvista ja lisäsin kääntäjälle yksikkötestit. Jotta testejä olisi helppo kirjoittaa, toteutin ne GLSL-lähdekooditiedostoina, joihin on kommentteina lisätty odotettu tuloste tai virhe. Aluksi kirjoitin parikymmentä testiä, mutta nyt niitä on 97. Testit auttoivatkin useita kertoja havaitsemaan ja korjaamaan kääntäjään livahtaneita virheitä. Ja toisinaan kääntäjään lisätyt virheentarkistukset paljastivat syntaksivirheitä testeissä.

Kun kaikki valmistelevat muutokset oli tehty, syntaksipuun muuttaminen SPIR-V-binääriksi oli lopulta varsin suoraviivaista. Dokumentaatiota sai kyllä lukea olan takaa – käytössä on lähes 200 eri alkeisoperaatiota. Aritmeettisista operaatioista on eri versiot eri tyypeille, ja mukaan lasketaan myös tyyppien ja vakioiden määrittelyt.

Testatakseni tätä osaa lisäsin yksikkötesteihin toiminnon, jossa käännetty SPIR-V-välikoodi syötetään Khronosin julkaisemalle oikeellisuudentarkistustyökalulle. Tämä ei tosin takaa, että käännetty binääri toteuttaa saman logiikan kuin lähdekoodi, mutta suurin osa operaatioista on niin yksinkertaisia, että logiikkaa on vaikea tyriä. Silmukkarakenteisiin olin kuitenkin onnistunut tekemään vasta myöhemmin paljastuneen bugin, jonka takia silmukoista tuli ikuisia.

```
uniform Transform
{
 mat4 mvp;
};
layout(location=0) in vec4 position;
void main()
{
 gl_Position = mvp*position;
}
```




```
...
%main__ = OpFunction %void None %void__
 %18 = OpLabel
 %21 = OpAccessChain %_ptr_Uniform_mat4 %13 %int_0
 %22 = OpLoad %mat4 %21
 %23 = OpLoad %vec4 %position
 %24 = OpMatrixTimesVector %vec4 %22 %23
 %26 = OpAccessChain %_ptr_Output_vec4 %10 %int_0
 OpStore %26 %24
 OpReturn
 OpFunctionEnd
```

data myös pirstoutuu, koska yksityinen osa on varattava dynaamisesti.

En halunnut tarpeetonta dynaamista muistinvarausta tai tehdasfunktioita, ja käännösaikainen grafiikka-alustan valinta sopi minulle hyvin. Kehitin ratkaisun, joka on tavallaan nurin käännetty rajapinta: yhteinen luokka perii alustakohtaisen luokan. Käännösjärjestelmä valitsee mukaan otettavan taustakoodihakemiston ja asettaa otsikkotiedostojen hakupolun valittujen käännösasetusten perusteella. Jokainen toteutus käyttää tyyppialiasta tuodakseen luokat näkyviin samoilla nimillä. Tästä ei seuraa ristiriitaa, koska vain yksi toteutus kerrallaan voi olla käytössä.

Alustakohtaisten luokkien kaikki jäsenet ovat suojattuja, joten sovellukset eivät voi käyttää niitä suoraan. Yhteinen luokka pääsee alustakohtaisiin jäseniin käsiksi perinnän kautta, mutta usein myös alustakohtainen luokka tarvitsee yhteisten jäsenmuuttujien arvoja. Käytin tässäkin ystävämäärityksiä, koska kyseessä on tavallaan saman luokan kaksi eri osaa.

OpenGL-koodin siirto taustaluokkiin oli enimmäkseen varsin suoraviivaista, mutta pari kohtaa vaati vielä uudelleenjärjestelyä. Erityisesti `Program`-luokan monimutkaista reflektiologiikkaa ja sen käyttämiä tietorakenteita sai miettiä. Koska yhteisen luokan jäsenten käyttäminen alustakohtaisesta luokasta on hieman vaivalloista, siirsin reflektiotietorakenteen määrittelyn erilleen muusta luokasta.

Toinen huomionarvoinen loppuvaiheen muutos oli ikkunaan liittyvän ruutupuskurin hallinta. OpenGL:ssä se on osa kontekstia ja aina käytettävissä, joten `Framebuffer`-luokasta oli sitä varten globaali instanssi. Tämä oli ongelmallista jo senkin takia, että useampaa ikkunaa käytettäessä jokaisella pitäisi olla oma ruutupuskuri. Vulkanissa näiden ruutupuskurien luonti on ohjelman vastuulla, joten siirsin instanssin `WindowView`-luokan omistukseen.

Tähän asti olin käyttänyt OpenGL:n vakioarvoja omien vastaavien enumeraatioiden pohjana. Se ei enää käynyt päinsä, sillä Vulkanissa numeeriset arvot ovat erilaiset. Jokaiselle enumeraatiotyypille tarvittiin muunnosfunktio omasta vakioarvosta grafiikka-alustan arvoon ja joillekin myös toiseen suuntaan. Omien arvojen määrittelystä oli myös hyötyä, sillä pystyin koodaamaan esimerkiksi pikseliformaattien ominaisuudet niitä kuvaavien vakioiden bitteihin.

Vulkanin perusteet

Vihdoin ja viimein pääsin itse asiaan eli käsiksi Vulkan-rajapintaan. Kopioin OpenGL-taustakoodin ja muutin kaikki funktiot tyngiksi, jotka vain heittävät poikkeuksen. Tästä lähtökohdasta oli helppo alkaa tekemään Vulkan-toteutusta luokka ja funktio kerrallaan.

Aivan ensimmäiseksi Vulkan-rajapinta on alustettava luomalla instanssi (`VkInstance`). Se toimii porttina muuhun rajapintaan, ja useimmat funktiot vaativat kahvan joko instanssiin tai sen avulla luotuun alemman tason olioon kuten laitteeseen. Poikkeuksen muodostavat jotkin implementaation ominaisuuksien kyselyyn käytettävät funktiot, kuten laajennosten listaus.

OpenGL:n tapaan Vulkan tukee laajennoksia, mutta niiden käyttöä on erikseen pyydettävä instanssia tai laitetta luotaessa. Myös jotkin edistyneemmät ominaisuudet kuten geometriashaderit tai pakatut tekstuuriformaatit on erikseen määriteltävä otettavaksi käyttöön. Mikäli implementaatio ei tue pyydettäviä laajennoksia tai ominaisuuksia, alustus epäonnistuu. Tämä helpottaa vianmääritystä, jos käyttäjän laitteisto ei tue jotakin ohjelman käyttämää laajennosta.

Useimmat Vulkania käyttävät ohjelmat tarvitsevat ainakin kolmea eri laajennosta. `KHR_surface` mahdollistaa renderöidyn kuvan esittämisen käyttöjärjestelmän tarjoamalla piirtopinnalla, useimmiten ikkunassa. Se tarvitsee rinnalleen alustakohtaisen laajennoksen, kuten `KHR_xlib_surface`, `KHR_wayland_surface` tai `KHR_win32_surface`. Lisäksi tarvitaan vielä `KHR_swapchain`-laajennos, joka mahdollistaa piirtokomentojen kohdentamisen piirtopintaan.

Suositeltava laajennos on myös `EXT_debug_report`. Sen avulla ohjelma voi asettaa takaisinkutsufunktion vastaanottamaan vianmääritystietoja. `EXT_debug_utils`-laajennoksella puolestaan voi antaa grafiikkaolioille selkokieliset nimet helpottamaan viestien lukemista. Nimet näkyvät myös ulkoisissa vianmääritystyökaluissa kuten RenderDocissa (renderdoc.org).

Vulkan tukee myös aputoimintojen lisäämistä kerroksina ydinrajapinnan päälle. Erittäin hyödyllinen kerros on `KHRONOS_validation`, joka tarkistaa funktiokutsut mahdollisten virheiden varalta. Rajapinnan monimutkaisuuden takia se on aloittelijalle lähes pakollinen ja nopeuttaa merkittävästi ongelmakohtien löytämistä myös edistyneemmille ohjelmoijille. Virheilmoitukset välitetään ohjelmalle edellä mainitun takaisinkutsun kautta.

Instanssin luomisen jälkeen seuraavaksi pitää saada kahva laitteeseen (`VkDevice`). Jos tietokoneessa on useampi näytönohjain (esimerkiksi sekä prosessoriin integroitu että erillinen kortti), sovelluksen on valittava niistä käyttöönsä sopiva. Myös useamman näytönohjaimen yhtäaikainen käyttö on mahdollista. Alkuun riittävä heuristiikka on valita ensimmäinen grafiikkaoperaatioita tukeva näytönohjain[2].

Siinä missä OpenGL tukee rajapinnan tasolla vain kaksoispuskurointia[3], Vulkanissa piirtopintaan kohdistuvia kuvapuskureita voi olla useampia. Usein käytetään kolmea puskuria: yksi on näkyvissä kuvaruudulla, toinen valmiina näytettäväksi ja kolmanteen piirretään seuraavaa kuvaa. Useamman puskurin käyttö voi tuottaa sulavampaa animaatiota mutta kasvattaa latenssia.

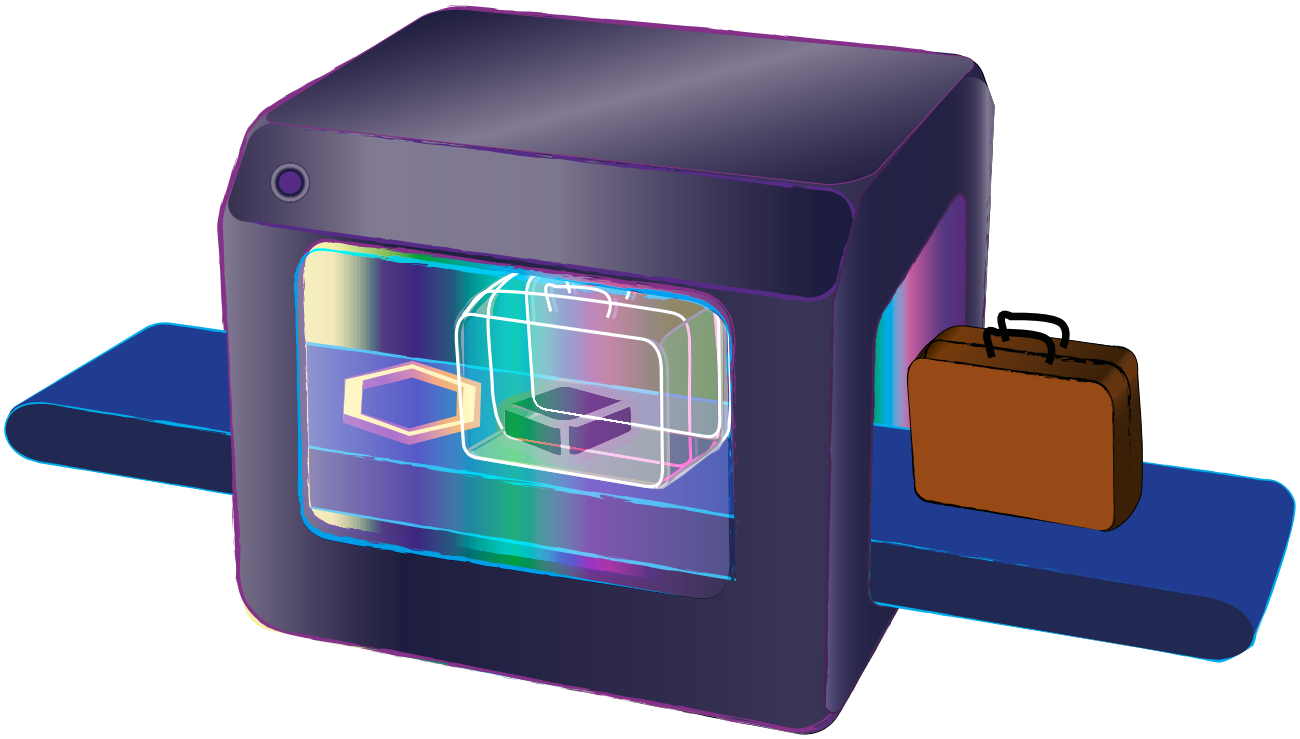
Kuvapuskureita hallitaan `VkSwapchain`-olion avulla. Swapchain sisältää yhden tai useamman kuvan, joita sovellus voi käyttää piirtokomentojen väripuskurina. Piirron aluksi pyydetään swapchainilta seuraavan vapaan kuvan indeksi, ja lopuksi valmis kuva lähetetään näytettäväksi.

Grafiikan renderöinti Vulkanissa jakautuu vaiheisiin (`VkRenderPass`). Kullekin renderöintivaiheelle on määritelty ruutupuskurin rakenne sekä siihen liitettyihin kuviin kohdistuvat toiminnot. Vaiheen alkaessa kunkin kohdekuvan sisältö voidaan ladata muistista, täyttää vakioarvolla tai jättää määritteleмättömään tilaan. Vastaavasti vaiheen loputtua sisältö voidaan säilöä muistiin tai heittää pois. Esimerkiksi syvyyspuskurin sisältöä tarvitaan usein vain renderöintivaiheen aikana.

Renderöintivaiheiden oikeaoppinen käyttö on tärkeää sulautetuilla laitteil-

[2] *Näytönohjain, joka ei tue grafiikkaoperaatioita, voi kuulostaa oudolta, mutta Vulkania on mahdollista käyttää myös pelkkään laskentaan.*

[3] *Jotkin ajurit toteuttavat kolmoispuskuroinnin myös OpenGL:ssä, mutta tämä ei näy sovellukselle.*



la, joiden grafiikkasuorittimet tuottavat kuvan pala kerrallaan. Suorittimen yhteydessä on pieni määrä hyvin nopeaa muistia, mutta päämuistin käyttö on hidasta. Turhia latauksia ja säilömisiä kannattaa välttää. Pöytäkoneiden näytönohjaimilla tämä ei ole niin merkityksellistä, koska nopeaa muistia on paljon.

Vulkanissa komennot koostetaan komentopuskuriin (`VkCommandBuffer`), joka sitten lähetetään grafiikkasuorittimelle. Komentopuskurit on sidottu tiettyyn komentojonoon, joita voi olla tarjolla useampia. Kaikki komentojonot eivät välttämättä tue kaikkia komentotyyppejä. Erillisten grafiikka- ja laskentajonojen käyttäminen voi myös edesauttaa näiden komentojen suoritusta asynkronisesti suhteessa toisiinsa.

Grafiikkakomentojen suorittamiseksi on ensin aloitettava renderöintivaihe `vkCmdBeginRenderPass`-komennolla. Komennolle annetaan renderöintivaiheen aikana käytettävä ruutupuskuri sekä tyhjennysarvot, jos jotkin kohdepuskurit on määritelty tyhjennettäviksi. Renderöintivaihe lopetetaan `vkCmdEndRenderPass`-komennolla.

Olin nyt saavuttanut ensimmäisen tärkeän virstanpylvään Vulkanin käytössä: grafiikkamoottorini pystyi näyttämään tyhjän ikkunan.

Vulkanin tuoreessa 1.3-versiossa mukaan otettiin `KHR_dynamic_rendering`-laajennos, jonka avulla voi välttää `VkRenderPass`- ja `VkFramebuffer`-olioiden käytön. Sen sijaan kaikki renderöintivaihetta koskeva informaatio annetaan `vkCmdBeginRendering`-komennolle. Haluan kuitenkin omassa grafiikkamoottorissani tukea myös mobiililaitteita, joten en ota laajennosta käyttöön ainakaan vielä.

Resurssit käyttöön

Seuraavaksi ikkunaan piti tietysti saada jotain sisältöä, joten ryhdyin toteuttamaan piirtokomentoja. Piirtoon käytettävä tila asetetaan linjasto-oliolla (`VkPipeline`), minkä lisäksi datalähteinä toimivat puskurit ja tekstuurit sidotaan erikseen. Linjastoa luotaessa shaderien SPIR-V-välikoodi käännetään grafiikkaprosessorin konekoodiksi, joka on verrattain aikaavievä operaatio. Sen takia samoja linjastoja kannattaa pyrkiä käyttämään uudelleen.

Koska linjasto sisältää monessa eri paikassa määriteltyä tilaa, uudelleenkäyttäminen on helpointa tehdä matalalla tasolla. Toteutin tätä varten välimuistin, joka pitää kirjaa jo luoduista linjasto-olioista. Kun linjaston tila on muuttuunut piirtokomentojen välillä, siitä lasketaan 64-bittinen tiiviste. Jos samalla tiivisteellä on jo luotu linjasto-olio, käytetään sitä, muutoin luodaan uusi.

Jotkin linjaston osat on mahdollista määritellä dynaamisiksi, jolloin niiden arvot asetetaan erillisillä komennoilla. Näin kannattaa tehdä etenkin piirto- ja leikkausalueille (*viewport*, *scissor*), jotta samaa linjastoa voi käyttää eri kokoisten ruutupuskurien kanssa.

OpenGL:stä tuttuun tapaan kolmioverkkojen kulmapisteiden arvot on ladattava puskuriin, jotta niitä voisi käyttää piirtokomennoissa. Ja kuten kaikki muukin Vulkanissa, myös puskurien hallinta on monimutkaisempaa. Sovelluksen vastuulla on osoittaa puskurille sopiva muistialue ja huolehtia datan siirtämisestä sinne.

Vulkan tarjoaa käytettäväksi useita eri muistityyppejä, jotka jakautuvat kolmeen pääkategoriaan: keskusmuisti, keskusprosessorille näkyvä grafiikkamuisti sekä vain grafiikkaprosessorin käytössä oleva muisti. Renderöinnissä tarvittavat resurssit kannattaa luonnollisesti sijoittaa grafiikkamuistiin, jotta niiden käyttö on nopeaa. Keskusprosessorille näkyvää grafiikkamuistia on suhteellisen vähän, yleensä 256 megatavua, mutta siellä sijaitsevan datan päivittäminen on nopeaa.

Muistialueita voi olla samanaikaisesti varattuna rajallinen määrä. Spesifikaatio lupaa rajan olevan vähintään 4096, mikä riittää kyllä kokeiluihin mutta loppuu oikeassa käytössä äk-

kiä kesken. Sovellusten onkin suositeltavaa varata muistia suurehko määrä kerralla ja sijoittaa useita resursseja saman muistialueen eri osiin. Omassa grafiikkamoottorissani valitsin allokaation oletuskooksi 1/256 koko grafiikkamuistista eli 6 gigatavun näytönohjaimellani noin 24 megatavua.

Kulmapistedataa ei yleensä päivitetä jatkuvasti, joten sitä ei kannata sijoittaa kallisarvoiseen keskusprosessorille näkyvään muistiin. Siitä tietysti seuraa, että sisältöä ei voi kirjoittaa puskuriin suoraan, vaan se on tehtävä välikäden kautta. Välikätenä toimii keskusmuistiin sijoitettu valmistelupuskuri (*staging buffer*), johon data ensin kirjoitetaan ja sitten kopioidaan lopulliseen paikkaansa.

Datan kopiointi puskurista toiseen on näytönohjaimella suoritettava komento, eikä komentojen suorittaminen milloin vain ole hyväksi suorituskyvylle. Niinpä kopiointipyynnöt kerätään jonoon ja komennot kirjataan komentopuskuriin seuraavan ruudun prosessoinnin alkaessa. Tähän tarkoitukseen toteuttamani `TransferQueue`-luokka huolehtii sekä valmistelupuskurien varaamisesta että komentojen suorittamisesta sopivaan aikaan.

Puskurien ohella myös kuvien muisti on varattava vastaavasti. Kuvien sisäinen esitysmuoto grafiikkamuistissa voi tehokkuussyistä olla jotain muuta kuin peräkkäisiä pikseleitä, joten sisältö asetetaan aina siirtojonon kautta. Myös mipmapit luodaan siirtojonon avulla.

Kuvia ei Vulkanissa voi käyttää juuri mihinkään sellaisenaan, vaan kuvasta on luotava näkymä (`VkImageView`). Jos kuvassa on mipmap-tasoja tai kerroksia (*layers*), näkymän voi rajata käyttämään vain osaa niistä. Samalle kuvalle voi olla useita eri näkymiä. Päätin kuitenkin olla tuomatta näkymiä oman grafiikkamoottorini rajapintaan ainakaan vielä, koska en keksinyt käyttötapausta, jossa siitä olisi erityistä hyötyä, ja ominaisuuden tukeminen kaikissa eri järjestelmissä olisi ollut hankalaa.

Kuvien käyttöön liittyy toinenkin erityispiirre. Parhaan suorituskyvyn saamiseksi eri operaatiot voivat vaatia pikselidatalta erilaista sisäistä rakennetta. Vulkanin termein kuvan on oltava operaatioon soveltuvassa asettelussa (*image layout*). Ei liene yllätys, että on sovelluksen vastuulla huolehtia asettelujen hallinnasta.

Shaderien käyttämiä resursseja ei Vulkanissa sidota yksitellen, vaan ne kootaan määrittelijäjoukoksi (*descriptor set*). Shaderilla voi olla käytössään useampi kuin yksi määrittelijäjoukko, jolloin resurssit voidaan jaotella esimerkiksi sen mukaan minkälaisista lähteistä ne tulevat. Se voi vähentää peräkkäisten piirtokomentojen välillä vaihdettavien resurssien määrää ja sitä kautta parantaa suorituskykyä.

Nykyisellään grafiikkamoottorini pitää määrittelijäjoukkoja välimuistissa samaan tapaan kuin linjastoja. Tällä menettelyllä on kuitenkin haittapuolensa. Ensinnäkin välimuistiin jää helposti vanhentuneita joukkoja, joiden viittaamista resursseista osa on tuhottu. Toiseksi joukot varataan poolista, jonka kapasiteetti määritellään resurssityyppikohtaisesti. Sopivan kapasiteetin määrittäminen etukäteen on hankalaa. Harkitsen lähestymistavan vaihtoa ja määrittelijäjoukkojen muuttamista ei-pysyviksi.

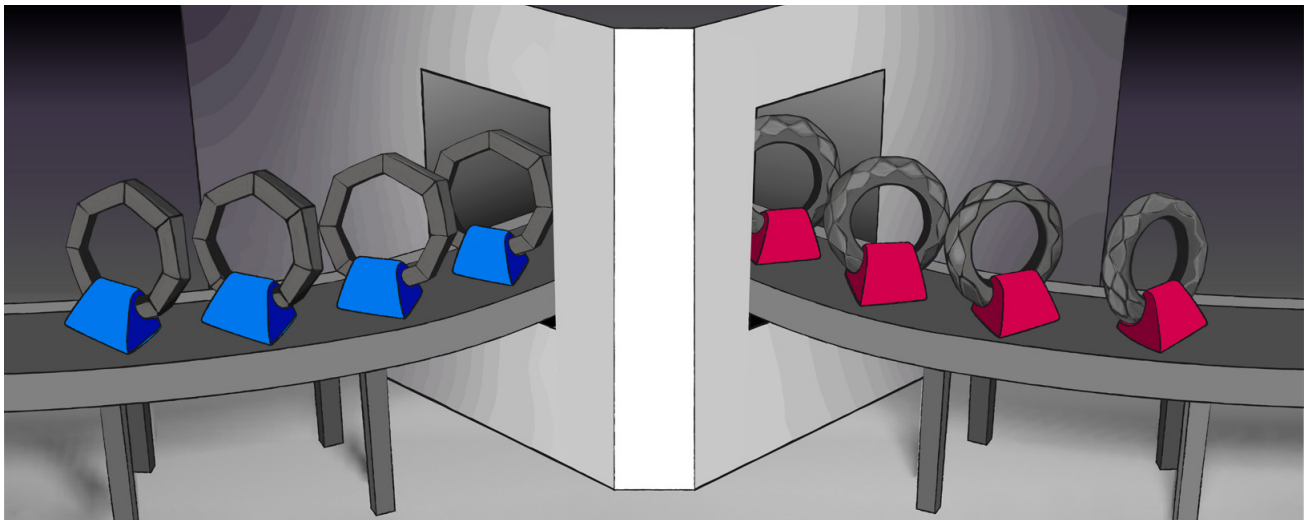
Huomionarvoisena yksityiskohtana Vulkanin koordinaatisto on hieman erilainen kuin OpenGL:ssä. Y-akselin nollakohta on ylhäällä ja koordinaatti kasvaa ruudun alareunaa kohti, päinvastoin kuin OpenGL:ssä. Kameraa lähinnä olevien pintojen syvyysarvo on 0, kun OpenGL:ssä se on -1. Nämä on helppo huomioida, jos koodaa pelkästään Vulkanille, mutta molempien rajapintojen tukeminen vaatii kikkailua.

Aluksi ajattelin ratkaista molemmat projektiomatriisia muuttamalla, mutta se aiheutti ongelmia monien efektien kanssa. Erityisesti varjoihin liittyvät efektit tekevät oletuksia sekä Y- että Z-akselista ja hajosivat muutoksen takia. Päädyin antamaan projektion Y-akselin olla ja käänsin kuvan ympäri aivan lopuksi sopivalla piirtoalueen määrityksellä. Ainoa tästä koituva haittapuoli on kuvan näkyminen ylösalaisin vianmääritystyökaluissa välivaiheita tarkastellessa.

Z-akseli oli hieman vaikeampi, koska sen arvot ovat grafiikkaprosessorille merkityksellisiä. Jonkin aikaa asiaa pohdittuani päätin hyödyntää omaa shaderikääntäjääni ja generoida tarvittaessa muunnoslauseen verteksishaderin loppuun. Julkisessa rajapinnassa näkyväksi käytännöksi valitsin Vulkanin nollasta alkavan arvoalueen, sillä se on yhdenmukainen syvyystekstuureihin tallennettujen arvojen kanssa.

Jonoon järjesty

Grafiikkasuoritin toimii asynkronisesti suhteessa pääsuorittimeen ja voi myös sisäisesti järjestellä komentoja uudelleen maksimoidakseen suoritusyksiköiden käyttöasteen. Vulkanin matalan abstraktiotason takia tämä



näkyy sovellukselle asti. Jos komentojen välillä on riippuvuuksia, on sovelluksen vastuulla varmistaa oikea suoritusjärjestys. Esimerkiksi kuvapuskuriin kohdistuvat piirtokomennot on suoritettava loppuun, ennen kuin kyseistä kuvaa käytetään tekstuurina.

Paitsi että komennot on suoritettava oikeassa järjestyksessä, on muistiin kirjoitettu data myös tehtävä näkyväksi sitä tarvitseville komennoille. Muistiesteillä (*memory barrier*) voidaan varmistaa, ettei välimuisteissa ole vanhentunutta dataa. Esteiden liiallinen käyttö voi aiheuttaa grafiikkasuorittimen sakkausta, joten loin `Synchronizer`-luokan pitämään kirjaa siitä, miten mitäkin resurssia on viimeksi käytetty. Kunkin renderöintivaiheen alkaessa tarkistetaan, onko jonkin resurssin käyttötapa muuttunut, ja lisätään tarvittaessa muistieste.

Muistiesteen yhteydessä voi myös muuttaa kuvien asettelua. Jos pikselidatan rakenne todellisuudessa muuttuu, grafiikkasuoritin joutuu siirtelemään sitä muistissa. Tarpeeton datan siirtely heikentää sekin suorituskykyä, joten asettelua kannattaa vaihtaa vain tarvittaessa. Jos esimerkiksi useampi peräkkäinen renderöintivaihe käyttää samaa ruutupuskuria, se kannattaa pitää piirtoon soveltuvassa asettelussa.

Muistieste on asetettava ennen renderöintivaiheen alkua, mutta käytettävät resurssit selviävät vasta renderöintivaiheen aikana. Kuinka siis luoda optimaalinen muistieste? Ratkaisin asian käyttämällä toissijaista komentopuskuria. Renderöintivaiheen sisältö kirjataan toissijaiseen puskuriin samalla, kun kerätään lista käytetyistä resursseista. Lopuksi kirjataan pääasialliseen komentopuskuriin muistieste ja kutsutaan toissijaista puskuria.

Jos keskussuorittimella ajettava koodi on kyllin nopeaa, kuten yksinkertaisissa ohjelmissa usein on, voi seuraavan ruudun valmistelu alkaa, kun grafiikkasuoritin vielä käsittelee edellisen komentoja. Resurssien muuttaminen niiden ollessa vielä käytössä johtaisi virheelliseen lopputulokseen. Jatkuvasti muutettavista resursseista täytyy olla useampi kopio – kolmoispuskurontia käytettäessä kolme kopiota. Sama koskee myös komentopuskureita.

Kopioiden ei tarvitse välttämättä olla itsenäisiä resursseja. Erityisesti uniform-lohkojen eri kopiot voi sijoittaa peräkkäin samaan puskuriin. Implementaatio voi vaatia lohkojen alkavan tietyn tavumäärän monikerran kohdalta. Käytän tätä lähestymistapaa omassa grafiikkamoottorissani.

Piirtokomennon tarvitsemaa tilaa asetettaessa on tietysti tiedettävä, mitä kopiota kulloinkin käytetään. Tähän oli jälleen tarjolla useita eri vaihtoehtoja. Lisäisinkö `Buffer`-luokkaan muuttujan ilmaisemaan aktiivista kopiota? Vai pitäisikö yksittäisten dataolioiden tietää, mikä kopio on käytössä? Entä mikä osa koodista olisi vastuussa kopion valitsemisesta?

Valitsin ratkaisun, jossa `Renderer`-olio pitää kirjaa nykyisen ruudun järjestysluvusta ja välittää sen erikseen sekä dataolioiden päivitysfunktiolle että linjaston sidontafunktiolle. Kun dataolion kaikki kopiot ovat samassa puskurissa, deskriptorijoukkoa sitoessa voi dynaamisella siirroksella määritellä, mitä kopiota sillä kertaa käytetään.

Grafiikkaprosessorin suoritusjonoon lähetettävien operaatioiden välillä voi myös olla riippuvuuksia. Ne synkronoidaan semaforeilla (`VkSemaphore`), jotka ovat erikseen luotavia synkronointioliota. Erityisesti swapchainin käsittelyn synkronointi on tärkeää. Ruutua varten pyydetty kuva ei välttämättä ole vapaana heti. Piirtokomentojen suorittaminen ennen kuvan vapautumista voisi aiheuttaa häiriöitä ruudulla näkyvään kuvaan, ja vastaavasti kuvan esittäminen liian aikaisin saattaisi tuoda näkyviin keskeneräisen kuvan.

Pitkät OpenGL-jäähyväiset

Grafiikkamoottorini tukee nyt muutamaa poikkeusta lukuunottamatta samoja ominaisuuksia sekä OpenGL- että Vulkan-rajapintaa käyttäen. Kalenteriaikaa projektiin meni reilu vuosi. Jatkossa aion käyttää ensisijaisesti Vulkania, mutta säilytän OpenGL-tuen vielä toistaiseksi. Myöhemmin saatan lisätä valikoimaan myös Direct3D 12:n ja Applen Metal-rajapinnan, mutta ne eivät ole korkealla tärkeysjärjestyksessä rajoitetun käyttöjärjestelmätuen vuoksi.

Kuten Skrollissa on jo aiemmin todettu, oman grafiikka- tai pelimoottorin tekeminen (ks. myös *Ylikuutio*-pelimoottori Skrolleissa 2018.4 ja 2019.2) on varsin hidas tapa tehdä pelejä. Pahimmillaan moottorin tekeminen vie kaiken ajan, eikä sitä pääse koskaan käyttämään pelien tekemiseen. Toisaalta kaiken tekeminen itse on kuitenkin opettavaista ja oman moottorin kanssa saatu kokemus on arvokasta myös valmiita moottoreita käyttäessä. Skenedemoissa oman grafiikkamoottorin käyttö on yleisempää kuin peleissä ja kokorajoitetuissa introissa jopa käytännössä pakollista.

Myös OpenGL on yhä riittävä moneen tarkoitukseen. Aloittelijalle se on helpompi oppia, ja useimmat modernin OpenGL:n käsitteet siirtyvät melko vaivattomasti Vulkaniin. OpenGL-tuki tuskin katoaa näytönohjainten ajureista vielä vuosiin, ja sen jälkeenkin OpenGL-ohjelmia on mahdollista ajaa yhteensopivuuskerroksen avulla.

Linkit

Vulkan-spesifikaatio:
vulkan.org/learn#key-resources

Virallinen ohjelmointiopas:
github.com/KhronosGroup/Vulkan-Guide
Kertoo rajapinnan käytöstä ja parhaista käytännöistä korkealla tasolla.

Tutoriaaleja:
vkguide.dev
Parempi tutoriaali, mutta sivuuttaa jotkin asiat käyttämällä ulkoisia kirjastoja.
vulkan-tutorial.com
Selittää kuinka kaikki tehdään itse, mutta sisältää joitakin kyseenalaisia käytäntöjä.

Kehitystyökalut:
vulkan.lunarg.com
Linux-jakeluista tämä löytyy yleensä paketoituna, esimerkiksi Debianissa libvulkan-dev.

Artikkelissa käsitellyn kirjoittajan grafiikkamoottorin lähdekoodi:
git.tdb.fi/libs/gl.git

Visualisoinnissa käytetyt esimerkkikoodit myös Skrollin verkkojatkoilla:
skrolli.fi/numerot

Tulossa pelimusiikki-tutkimusta

Suomen musiikkitieteellisen seuran (mtsnet.fi) vertaisarvioitu aikakauslehti *Musiikki* julkaisee tämän vuoden lopussa teemanumeron pelimusiikista. Teemanumeron kirjoituskutsu päättyi 28.2. Kutsusta käy ilmi, että teemanumeron tavoite on rakentaa kuvaa nimenomaan kotimaisesta pelimusiikkitutkimuksesta. Eri alojen tutkijoiden lisäksi myös kotimaisia pelimusiikin tekijöitä kutsutaan kirjoittamaan teemanumeroon, ja tutkimusartikkelien lisäksi muunkin tyyppiset tekstit ovat tervetulleita.

Pelimusiikki on kiinnostava pelikulttuurin osa-alue. Laaja-alainen, kotimainen lukupaketti pelimusiikista kuulostaa maallikkonäkökulmastakin erittäin lupaavalta. Jäämme odottamaan mielenkiinnolla. Teemanumero ilmestyy vuoden 2022 lopussa.

Eino Keskitalo

Musiikki-lehden irtonumeroita voi tiedustella seuralta osoitteen mtsnet.fi/musiikki *kautta.*

Pelitutkimuksen vuosikirja 2021

Suomen pelitutkimusseura (pelitutkimus.fi) on julkaissut joulukuussa *Pelitutkimuksen vuosikirjan 2021*. Kyseessä on vuosittainen verkkojulkaisu, jossa julkaistaan suomeksi leikki- ja pelitutkimusta. Pelitutkimuksen vuosikirja on avoin, vertaisarvioitu tiedejulkaisu, ja sitä on julkaistu säännöllisesti vuodesta 2009 lähtien.

Vuosikirjassa on erityyppisiä artikkeleita: vertaisarvioituja tutkimusartikkeleita, katsauksia eri tutkimuskysymyksiin, essee, kirjallisuusarvioita sekä peliaiheisten väitöstutkimusten lektioita. Aiheita on moninaisia, kuten mölkkymäiseen babka-peliin liittyviä arkeologisia löydöksiä, roolipelisessioiden suoratoistamista ja suomalaisen mobiilipeliteollisuuden käsittelyä Pelit-lehdessä. Juttujen silmäilemisen seurauksena päätyy omissa jatkotutkimuksissaan helposti kaninkoloon jos toiseenkin!

Vuoden 2022 vuosikirjan kirjoituskutsu on tämän Skrollin ilmestymisen aikaan vielä avoinna. Takaraja teksteille riippuu tekstityypistä.

Eino Keskitalo

Pelitutkimuksen vuosikirja 2021: pelitutkimus.journal.fi/issue/view/7755
Kirjoituskutsu vuosikirjaan 2022: pelitutkimus.journal.fi/announcement/view/414

Vertaisarvioitu artikkeli suomalaisista panopeleistä

Historian valossa millä tahansa medialla, luolamaalauksista vessanseinäkirjoituksiin ja mustavalkovalokuvista Usenetin uutisryhmiin, pornografisen sisällön olemassaolo on sääntö eikä poikkeus. Sama pätee myös tietokonepeleihin mediana. Aihetta pureskellaan kattavasti **Susanna Paasosen** ja **Veli-Matti Karhulahden** joulukuussa 2021 ilmestyneessä artikkelissa *Finnish Fuck Games: A Lost Historical Footnote* eli suomeksi kutakuinkin ”suomalaiset panopelit – kadonnut historiallinen alaviite”.

Eksakti käsitteiden määrittely on olennainen osa tiedettä. Panopelin tutkijat määrittelevät pelattavaksi ohjelmaksi, joka tyypillisesti sisältää tai simuloi yhdyntää. Vakava tutkimusartikkeli vähemmän vakavasta aiheesta tarjoaa faktasisällön lisäksi viihdyttävän lukuelämyksen.

Artikkeli luo lyhyen katsauksen panopelien yleiseen historiaan, minkä jälkeen se keskittyy analysoimaan neljää kotimaista panopeliä: *Striptease Ventti*, *Helttaa Helmaan*, *Bepa Quest* sekä *Koulu3*. Tutkimusmenetelmänä on historiatietoinen tulkinnallinen pelaamisen tutkimus, eli pelejä on pelattu ja luettu ja ne on pyritty sijoittamaan julkaisuaikakautensa kontekstiin.

Pelejä on verrattu myös muihin aikakauden vastaavan kaltaisiin peleihin. Artikkelin lopussa tutkijat toteavat, että aikaisemman panopelitutkimustiedon puutteen takia suomalaisesta panopeliskenestä on mahdoton sanoa, onko se ainutlaatuinen vai kenties tyypillinen. Jatkotutkimusta tee-se-itse-panopeleistä siis tarvitaan. Sitä odotellessa suosittelemme Skrollin artikkeleita *Digitaalisen erotiikan varhaisvuodet* numerossa 2013.3 sekä *Pornopelit* 2015.1:ssä (pdf-lehdet maksutta: skrolli.fi/numerot, irtonumeroita paperilla: skrolli.fi/kauppa).

Vesa Linja-aho

Lopullinen taitettu artikkeli löytyy lähteestä maksumuurin takaa. Turun yliopiston tutkimusportaalissa on myös maksuton vedostallenne osoitteessa research.utu.fi/converis/portal/detail/Publication/68184635. *Lähde: Paasonen, S., & Karhulahti, V. (2021). Finnish Fuck Games: A Lost Historical Footnote. In V. Navarro-Remesal & Ó Pérez-Latorre (Eds.), Perspectives on the European Videogame (pp. 75-90). Bristol University Press. doi:10.1017/9789048550623.005.*

Figure 2. Strip-tease Ventti. [That's ok, I am Ulla.]

***Helttaa Helmaan* (“Egg”)**

Also designed for Commodore64, *Helttaa Helmaan* reminds one of a classic arcade game such as *Space Invaders* (1978) and *Galaxian* (1979). Nevertheless, instead of a spaceship, the player is in control of a giant penis (named Kyrpä-Kari i.e. “Kari the Dick”) with the goal to maintain erection and – instead of shooting aliens – hitting and penetrating diverse orifices until the level of arousal drains off. The title's time of release differs depending on the source, most referencing a year between 1985 and 1988. This fuck game is also signed by an artist, “The Incredible Mr. Eggman”, which links to the product's English subtitle, *The Egg*. This is either an intentional or unintentional pun of literally translating the Finnish word “muna” that refers diversely to a testicle, a penis, and an egg.

Helttaa Heimaan follows conventionally the stereotypes of gaming avatars by providing the player with a somewhat masculine male character to control (see Ivory 2006; Miller & Summers 2007). In comparison to these stereotypes, Kari is also very much sexualized and not merely in a heteronormative

Kati Alhan väitöskirjatutkimukseen perustuva artikkeli Kun ilmaispelit valloittivat maailman tässä Skrollissa s. 74–77.

Devaajat, devaajat, devaajat!

Sultant of Con kalibroi ohjelmistokehittäjien odotukset ja prioriteetit uusiksi.

En ole aivan tarpeeksi vanha pukumieheksi, mutta olen todistanut IT-alan muutosta kauluspaitainsinöörien ajasta nykyvaiheeseen, jossa *Hobo or Dev?* -kisailu on "hauskaa" lounasajanvietettä Helsingin boheemikortteleissa. Arveluttavassa kisassa arvuutellaan, onko vastaantulija pummi vai ohjelmistokehittäjä, koska molempien oletetaan nykyisin olevan resuisen näköisiä. Liekö tämä monia variaatioita nähnyt peli on saanut alkunsa **Claire Lachow'n** *Hobo or Hipster?* -piirroksesta vuodelta 2009. Kuvassa on otsikon alla parrakas pipoon ja ruudulliseen flanellipaitaan pukeutunut hahmo.

IT-maailman kiertokulku on versonut uuden sukupolven devaajia, ja uudessa normaalissa riittää kaikilla ihmeteltävää. Näin myös Sultant of Boomersilla, jonka ikänäköisissä silmissä jokainen päivä toimistolla näyttää Lachow'n taiteelta. Miksi joka toisella devaajalla on parta? Voisiko IT-yrityksissä olla jonkinlainen partakiintiö? Ja jos kauluspaitojen palauttaminen on liian radikaalia, **Steve Jobsin** ideaa yritysunivormusta kannattaisi harkita uudelleen: yksiväriset poolopaidat ruutupaitojen tilalle.

Pinnallisten havaintojen takana piilottelee kuitenkin myös oikeita ongelmia.

Epärealistiset odotukset

Nuoret ovat aina olleet nuoria ja muodit vaihtelevat. Ainoa vakio on, että kirkasotsaisetkin devaajat paatuvat vuosien saatossa kyynisiksi kolumnisteiksi. Silti joudun säännöllisesti hämmästelemään, kuinka maailmasta vieraantuneita asenteita korkeakoulujen kursseilta päästetään lävitse. Perusteet ja alkeellinenkin tietotekniikan historian tuntemus tuntuvat olevan hukassa, minkä seurauksena ennakko-odotukset käytännön projekteista ovat pahasti pielessä. Jos sinulla on asemasi puolesta mahdollisuus vaikuttaa nuorten kehittäjien koulutukseen, kuule hätähuutoni.

Olen ollut todistamassa aivan liian montaa naamapalmuhetkeä, jossa laumallinen junioridevaajia on innosta puhkuen tullut mukaan asiakasprojektiin. Sitten kun kuvun alta on paljastunut valtava läjä vanhaa koodia, kauhistus ja pettymys on ollut käsin kosketeltavaa. Kun "enterpriseä" lähdetään perkaamaan, modernista pilvinatiivista ei nimittäin usein ole tietoakaan. Myönnän ettei tällaisen perinnekoodin tunkkaaminen ole mieltä ylentävää konkarillekaan, mutta jostain on aina aloitettava – ja parempi aloittaa tosimaailmasta kuin haavemaailmasta. Fakta on, että IT-alan roolina on ylläpitää uuden ja kauniin lisäksi myös vuosia ja vuosikymmeniä vanhaa teknologiaa. Sekin on tärkeää. Välillä se on todella tärkeää.

Myös nykydevaajissa esiintyvä korostunut arvostus näennäisen "merkityksellisiä" tehtäviä kohtaan herättää vanhassa kon-sultissa ihmetystä. Eettisen vegaaniravintolan mobiilisovelluksen kehittämisestä saattaa saada enemmän maailmanhalailuviboja someensa kuin rautakaupan varastonhallinnasta, mutta työn luonteesta tai merkittävyydestä tämä ei vielä kerro paljoakaan. Haastavimpia ohjelmistokehitystehtäviä sekä yhteiskunnallisesti merkityksellisempää infrastruktuuria voi löytyä näennäisen tylsiltäkin sektoreilta. IT:tä on kaikkialla, eikä lopulta mikään toimisi ilman sitä.

Tärkeysjärjestys kuntoon

Mikä pahinta, näillä pinnaltaan "merkityksellisillä" tehtävillä voidaan suoraan sanottuna sumuttaa työntekijöitä. Eettisillä tai kivoilla hommilla voidaan yrittää perustella sitä, miksi tekijöille maksetaan vähemmän palkkaa tai bonuksia kuin reilu talousmatematiikka edellyttäisi. Tällä kaikkea muuta kuin eettisellä kikalla saatetaankin sitten kirittää firma nopeammin pörssikuntoon.

Konttoreilla sama ilmiö näkyy siinä, että varustelutasoa nostetaan näennäisesti. Enää ei riitä, että toimistolla on Moccamaster tai asiallinen kahviautomaatti. Niiden tilalla on oltava kymppitonnin keittimet tai ihka aito barista tekemässä kofeiinittomia lattemachhiatoja devaajille. Tämän kahvipesun jälkeen onkin sitten helppo unohtaa, että vaikka projektin kohde olisikin mukava ja juomaa riittää, itse koodaustyö voi olla aivan samaa tai pahempaa kuin jonkin konservatiivisen konepajayhtiön enterprise-projektissa. Yhtä lailla joku myös käärii tuotot molemmista – jos et sinä, niin sitten joku muu.

Uusiin tehtäviin kannattaa suhtautua uteliaisuudella ja avoimella mielellä, mutta itseään ei myöskään pidä myydä liian halvalla. Seuraavan kerran kun joku yrittää kaupata sinulle huonoa diiliä sulosoinnuilla ja ilmaisilla virvokkeilla, Sultant of Con määrää lääkkeeksi pätkän vanhojen ihmisten sketsiohjelmaa: *Ketosen ja Myllyrinteen* **Pasin** (youtu.be/8JBT2Q-cJfI).

"Ahersit yötä päivää pelkän kolajuoman voimalla vain, että sinua arvostettaisiin. Se on hienoa, Pasi."

"Sinun kuuden kuukauden koeaikasi on nyt päättynyt ja se ohjelma on valmis. Nyt setä alkaa käärimään sillä fyffeä."

"Joten painu Pasi…"

Devaamisiin!

Sultant of Con

P.S. Minulla ei ole ollut partaa sitten vuoden 1995, mutta kun sellainen oli, se kävi varoittavasta esimerkistä: tuohon aikaan "soul patch" oli kova juttu, eikä se ole ikääntynyt sen paremmin kuin muutkaan partamuodit. Onnekseni sosiaalinen media rajoittui IRCin tekstiin ja paperikuvat on poltettu.

Paperiton tulevaisuus. Nyt?

Janne Sirén

Sirénin sisäpiiri -palstalla Janne Sirén ruotii seuraamiaan teknologiailmiöitä ja -alakulttuureita. Tällä kertaa tarkastelussa on siirtymiä paperilta sähköiseen.

Tulevaisuuden toimisto (*The Office of the Future*) – näin mahtipontisesti otsikoi Business Week -lehti sittemmin ahkerasti siteeratun artikkelinsa, joka ilmestyi 30. kesäkuuta 1975. Jutun ensimmäinen väliotsikko lupasi vieläkin enemmän: paperittoman toimiston (*The Paperless Office*). Peräti paperikopiokoneistaan tuttu Xerox, tutkimuskeskusjohtajansa **George E. Paken** edustamana, oli saatu mukaan uumoilemaan paperin kuolemaa.

"Ei ole epäilystäkään siitä, että seuraavan 20 vuoden aikana toimistot kokevat vallankumouksen", Pake ennakoi. Vuonna 1995 hänen toimistonsa olisi täysin erilainen – työpöydällä olisi jotain, jota Pake kutsui televisioterminaaliksi: "Voin hakea dokumentteja arkistoistani ruudulle napin painalluksella… Voin hakea postini ja kaikki saamani viestit. En tiedä, kuinka paljon painomateriaalia kaipaisin tällaisessa maailmassa."

Vuosikymmen ennen Paken kommentteja **J. C. R. Licklider**, amerikkalainen psykologi ja tietojenkäsittelytieteilijä, arvioi elektronisen julkaisemisen tulevaisuutta *Libraries of the Future* -kirjassaan. Hän siteerasi moderniksi maksiimiksi kutsumaansa sanontaa: "Ihmisillä on tapana yliarvioida mitä voidaan tehdä vuodessa, mutta aliarvioida, mikä on mahdollista viidessä tai kymmenessä vuodessa." Sanontaa on sittemmin soviteltu niin **Bill Gatesin** kuin **Arthur C. Clarkenkin** suuhun, mutta tosiasiassa sen alkuperä taitaa olla tuntematon.

En tiennyt, mitä kummastakaan sitaatista pitäisi ajatella, kun reilu 20 vuotta myöhemmin ryhdyin testaamaan IBM ThinkPad TransNote -nimistä kannettavaa tietokonetta MikroBitti-lehdelle. TransNoten pääominaisuus oli integroitu paperilehtiö, johon tehdyt merkinnät kopioituivat automaattisesti tietokoneelle.

IBM TransNote

Vuoden 2001 maaliskuussa kirjoittamani TransNote-testi julkaistiin MikroBitin numerossa 5/2001. Paperilehtien arkisto kertoo minun kiteyttäneen tuotteen seuraavasti: "IBM ThinkPad TransNote on mielenkiintoinen askel tuntemattomaan. Se toimii riittävän hyvin ollakseen vakavasti otettava yritys muuttaa tottumuksiamme, joskin muistilehtiön käsittelyn kömpelyys ja laitteen korkea hinta heikentävät sen houkuttelevuutta." Sähköinen arkistoni vahvistaa, ettei sitaattia ole muutettu lehden toimituksessa. Laitteen hinnaksi ilmoitettiin hulppeat 25 000 markkaa (nykyrahassa 5 500 €).

Sen sijaan tämä käsien heilutteluni oli tiputettu lehdestä pois: "Hyvistäkään muistiinpanoista ei ole mitään hyötyä, jos niitä ei löydy silloin kun

Päivitys: Suomen postikaaos

Digiloikan odottamaton vauhdittaja

Mitä sanoo GDPR, kun henkilötiedot hukkuvat fyysisen postipaketin mukana?

Helmikuun alussa Suomessa uutisoitiin Postin virheestä, jonka seurauksena Kelan Kouvolan palvelupisteeseen jätettyjä asiakirjoja katosi postimatkalla laitoksen skannauskeskukseen. Seurantakoodin mukaan Posti oli ottanut paketin vastaan, mutta siihen jälki sitten kylmenikin. Vuotta aikaisemmin Posti oli otsikoissa Suomen Kansallisarkiston kanssa, kun jälkimmäinen ilmoitti lopettavansa postin käyttämisen kaukolainoihin. Arkiston ainutkertaisia aineistoja oli toistuvasti toimitettu väärään paikkoihin tai hukattu kuukausiksi.

Skrollissa ongelmamme Postin kanssa alkoivat kesän 2019 postilakkojen jälkeen. Postin kulku ei nähdäkseni koskaan toipunut lakoista tai niiden yhteydessä syntyneistä toimintamuutoksista. Tätä ennen Skrollit jaettiin pääasiassa päivän, parin tarkkuudella ilmestymispäivänään. Vaikka Skrollille sovitaan edelleenkin Postin kanssa ilmestymispäivä (nykyisin neljä keskiviikkoa maalis-, kesä-, syys- ja joulukuussa), vuoden 2019 kesän jälkeen lehdet on jaettu reilun viikon aikana, joka alkaa karkeasti paria päivää ennen valittua "ilmestymispäivää" ja jatkuu siitä noin viikon verran eteenpäin. Joskus, kuten Skrolli 2021.4:n kohdalla, tämä viikko venyy useammaksikin viikoksi.

Luovuimmekin jo vuonna 2020 perinteisestä perjantain ilmestymispäivästämme, koska sitä seurannut viikonloppu vain pahensi ja-

5. vuosikerta

Tämän Sirénin sisäpiirin myötä palsta on seurannut erilaisten teknologiailmiöiden ja -alakulttuurien menoa yhtäjaksoisesti jo viiden Skrolli-vuosikerran osana, alkaen numerosta 2018.3. Kiitos kaikille, jotka ovat jaksaneet mukana! Menneet aiheeni ovat jakautuneet seuraavasti:

- Commodore/Spectrum: 50 % (7)
- Keinotodellisuus: 29 % (4)
- Robotiikka: 14 % (2)
- Survivalismi: 7 % (1)

Kuten palstan esikuvat vanhoissa tietokonelehdissä, joiden sisällöt vaihtelivat esimerkiksi uutisista ohjelmointiin, myös tämän sisäpiirin tarkoitus on sekä kuvata tekijänsä kiinnostuksen kohteita että tuottaa lukijoille vaihtelevia näköaloja. Vaikka uutiset vanhojen tietokoneiden ja keinotodellisuudesta ovat muodostaneet palstan selkärangan, maailmassa on muutakin.

Laajennan tänäkin vuonna aiheita sekä tuttujen että ajankohtaistenkin teemojen ulkopuolelle. Ensimmäisenä ponnistan viime Skrollissa Commodore/Spectrum-sisäpiirin yhteydessä käsitellystä retrotuotteiden postituskeskustelusta laajempaan digiloikkaan. Kuten arvata saattaa, en tosin täysin kykene välttelemään vanhoja koneita tälläkään kertaa.

Vanhat Sirénin sisäpiirit löytää helpoiten Skrollin suuren sisällysluettelon kautta (skrolli.fi/sisallysluettelo/palstat). Noin vuotta vanhemmat pdf-lehtemme ovat myös ladattavissa maksutta samasta paikasta. Palstalle voi kirjoittaa osoitteeseen: janne@skrolli.fi.

Kuva: Janne Sirén

Tulevaisuus on kädessäsi. Ainakin jos kuvassa olevaa Amiga-kuulakärkikynää on uskominen, sillä siinä lukee "the future is in your hand". Kynän alla vasemmalla on tavallinen Moleskine-paperikirja. Vasemmalla takana on IBM ThinkPad TransNote -muistikirjatietokone, keskellä reMarkable 2 -sähkömustetabletti ja oikealla etualalla Moleskine Smart -älykynä ja sen "paperitabletti".

niitä tarvittaisiin – ja liian usein paperit ovat hukassa. Lisäksi paperi on nykyisenkaltaisessa mobiilissa maailmassa hankala, fyysisten ominaisuuksiensa rajoittama media. Paperisten muistiinpanojen välittäminen maapallon toiselle puolelle vaatii niiden siirtämistä tietokoneelle käsin, skannaamista tai hidasta kuriiripalvelua. Paitsi jos sattuu omistamaan TransNoten." En varsinaisesti moiti toimitusta tästä.

IBM ilmoitti lopettavansa TransNoten myynnin seuraavan vuoden helmikuussa, ja vuosikymmenen puolivälissä yhtiö myi koko ThinkPad-tuotantonsa Lenovolle. TransNoten ongelma ei kuitenkaan ollut sen toteutuksessa. Vaikka jotkut arvioitsijat olisivat kaivanneet Pentium III (600 MHz) -pohjaiseen kannettavaan lisää tehoja, kone voitti vuoden mittaisen elinkaarensa aikana myös useita palkintoja, kuten teollisen muotoilun iF- ja IDSA-palkinnot sekä PC Magazine -lehden parhaan teknologian palkinnon PC-kategoriassa. Ihan aiheesta.

Kaksikymmentä vuotta myöhemmin kirjoitan edelleen paperilehteen, tällä kertaa Skrolliin. Vieressäni oleva TransNote näyttää jotakuinkin yhtä futuris-

keluviivettä. Eipä silti, sittemmin Posti on myös lakannut jakamasta kirjepostia perjantaisin, tiistaisin ja joillain alueilla vuoroviikoin kaikkina muinakin viikonpäivinä. Keskiviikko taitaa silti edelleen olla jakelupäivistä paras kompromissi, koska tällöin lehdellä on edes pieni mahdollisuus saapua suurimmalle osalle tilaajista yhden arkiviikon sisällä.

Olemme joutuneet luopumaan myös tavastamme lähettää pdf-lehdet tilaajille vasta, kun paperilehti on saapunut jotakuinkin kaikille. Viive olisi aivan liian pitkä. Varhain lähetetty pdf-lehti tarjoaa myös kaikille yhtäläisesti luettavaa.

Suomen pakettikaaos

Kokemukseni mukaan Postin pakettikuljetus ja kansainvälinen liikenne pärjäsivät jonkin aikaa lehtijakelua paremmin, lukuun ottamatta ymmärrettävää pandemiahäiriötä keväällä 2020. Viimeistään kesän 2021 tullaussäännösten laajentaminen pakottamaan myös alle 22 euron arvoiset EU:n ulkopuoliset lähetykset – mukaan lukien kirjeet – tullattaviksi, tuntuu kuitenkin ajaneen tämänkin puolen kaaokseen. Posti on julkisuudessa kannattanut uudistusta muun muassa runsaan Kiina-liikenteen aiheuttamien kustannusten takia, joten ei-toivotusta viranomaisratkaisusta ei tunnu olevan kysymys. Lähetyksistä nimittäin peritään Postille maksettava 2,90 € käsittelymaksu.

Käsittelin uudistuksen seurauksia edellisen Sirénin sisäpiirin laatikossa *Suomen postikaaos* (Skrolli 2021.4); järjestelmä ei toimi luvatusti ja ennen viikon matkalla ollut kirjeliikenne on hidastunut kuukaudella. Kirjoitukseni ja samassa lehdessä julkaistu **Mikko Heinosen** *Ei näin!* -artikkeli logistiikan murroksesta herättivät vastakaikua lukijoissa. Esimerkiksi nimimerkki **juj** viestitti, lyhennettynä: "Postikaaosartikkeli oli hyvä. Tuotti suurta helpotusta lukea sitä, kun omat posti- ja tullikokemukset ovat olleet tänä vuonna yhtä hajoilemista ja henkisen tyyneyden koetteluharjoituksia. 'Posti on noudettavissa noutopisteessä koodia vastaan' -episodi on [myös] tullut koettua monta kertaa eli 'koodi on lähetetty sähköpostiin', jota ei lähetykseen ole rekisteröity ollenkaan."

Nimimerkki **RetroBard** puolestaan kommentoi Postin keräämää käsittelymaksua, jota ei makseta niin kutsutuista IOSS-lähetyksistä, joiden ALV on maksettu jo lähtömaassa: "Ei voi olla huomaamatta, kuinka helpoksi käsittelymaksun maksaminen on tehty verrattuna siihen kuinka vanhanaikainen ja suoraan sanoen brezhneviläinen prosessi käsittelymaksun vapautuksen anominen on. Jos kerran Posti ja ilmeisesti myös Tulli tietävät, että ALV on maksettu lähtöpäässä, niin miksi ihmeessä sitä enää Suomen päässä tarvitsee sen tarkemmin ruveta selvittämään?" Ilmeisesti moni muukin oli tähän ilmeiseen epäkohtaan tyytymätön, sillä Posti tuli joulukuussa sen verran vastaan, että tällaiset lähetykset vapautuivat automaattisesti Postin käsittelymaksun maksamiselta.

RetroBard kommentoi myös tois-

tiselta kuin vuosituhannen vaihteessakin. Taittuvan nahkakansion sisään rakennettu läppärin ja digitaalisella lukijalla varustetun lehtiötelineen yhdistelmä ei tunnu kovinkaan paljoa menneisyyden reliikiltä. Tämä ei johdu vain siitä, että materiaaleiksi valittu musta (keino)nahka ja muovi ovat sangen ajattomia – tässä on jotain muutakin.

Paperipuolella laitteelle kerrotaan pienen lisäruudun kautta, mille lehtiön sivulle ollaan tekemässä merkintöjä. Kirjoittamiseen käytetään tietokoneen mukana tulevaa kynää, joka saa virtansa pienestä sormiparistosta. Muuten TransNoten lehtiölle kirjoitetaan kuin mille tahansa paperille, ja käytetty lehtiökin voi olla mitä tahansa paperia. Koneessa on jopa pari lokeroa mustepatruunoille. Paperille tehdyt merkinnät tallentuvat automaattisesti tietokoneelle kuvina, ja luonnollisesti myös lehtiön paperit mustejälkineen voi säilyttää, joko lehtiössä tai irti repäisten. Itse asiassa digitaalinen paperinlukija toimii silloinkin, kun tietokone on poissa päältä. Lukijassa on tätä tarkoitusta varten muutaman megatavun flash-muisti.

Windows 2000 -pohjaisen TransNoten saranoille nouseva ja eri asentoihin kääntyilevä näyttö muistuttaa moderneja hybriditietokoneita, vaikka edelsi niitä yli vuosikymmenellä. Konetta voi käyttää vaikkapa niin, että käyttäjä piirtää paperipuolelle samalla, kun näyttö näyttää kuvaa piirroksesta vastapäätä istuvalle yleisölle. TransNotessa on jopa kosketusnäyttö, jota voi käyttää pysty- tai vaakatasossa, sekä fyysisen näppäimistön lisäksi myös virtuaalinäppäimistö. Täyssähköisiä muistiinpanoja voi tehdä lisäksi paperilehtiön alta löytyvällä piirtopöydällä – TransNoten kuulakärkikynässä on osoitinkynä (*stylus*) toisessa päässä. Kaiken kukkuraksi koneen nahkakansio taittuu niin, että sekä paperilehtiö että tietokone kääntyvät tarvittaessa toistensa alle piiloon.

Kaikki tämä toimii oikein intuitiivisesti, lukuun ottamatta ehkä sitä manuaalisesti tehtävää sivunvaihtoa ja tekstintunnistuksen puutetta (tunnistus löytyy vain avainsanatoiminnolle), mutta nämä olivat aikakautensa rajoja. Laitteen perusoivalluskin on osoittautunut oikeaksi: Käsin tehdyt muistiinpanot ovat kasvotusten tapahtuvissa kokouksissa edelleen suotavampia ja sujuvampia kuin kannettavan naputtaminen. 2010-luvun lopulla Princetonin yliopistossa tehdyssä tutkimuksessa todettiin käsin kirjoittamisen jopa edesauttavan muistamista.

Kuva: Janne Sirén

TransNoten piirtoalustaa operoidaan tökkimällä kynällä laidan syvennyksiä. Niillä valitaan muun muassa paperia vastaava sivunumero. Alustan tila näkyy pieneltä näytöltä.

TransNotea vaivasi tosin sama ongelma kuin nykyistä puettavaakin tekniikkaa: ihmiset ovat erilaisia. Tässä tapauksessa sama laite ei oikein sopinut oikea- ja vasenkätisille. IBM vastasi huutoon – artikkelin alkukuvissa näkyy harvinaisempi vasenkätisten malli

Kuva: Janne Sirén

Viimeistään joulun jälkeen postilaatikossa odottanut nippu uusia tullauskirjeitä sai epäilemään tullausjärjestelmän toimivuutta ja postin käytön mielekkyyttä. Kuudesta kirjeestä peräti neljä sisälsi etukäteen maksetun IOSS-lähetyksen tullauspyynnön, joiden IOSS-numero oli myös toimitettu Postille ja joiden tulli-ilmoituksesta Postin olisi pitänyt huolehtia – kymmenistä saamistani lähetyksistä vain kerran tai kaksi Posti on hoitanut tämän. Viidennessä kirjeessäkin oli esimaksettu IOSS-lähetys, jonka IOSS-numerokaan ei ollut jostain syystä välittynyt Postille saakka. Kaikki lähetykset tulivat Englannista; paljon puhuttaneesta Kiina-postista ei siis ole kohdallani kysymys.

ta hieman varhaisempaa parannusta: "Postikaaosartikkeli oli kyllä aivan loistava. Jaan tämä tuskan aivan täysin. Jotain muutosta on tapahtunut tuon Postin käsittelymaksun suhteen, kun joissain tapauksissa tullaus on mahdollista tehdä appin kautta ennakkoon, jolloin käsittelymaksu on 'vaivaiset' 90 senttiä."

Tämä on totta. Itse olen tosin saanut tilaisuuden tullata lähetyksen mobiilisovelluksessa vain kerran, ja sekin kaatui siihen, ettei lähetys olisi saanut olla yli 150 euron arvoinen.

Tullaus postin mobiilisovelluksessa on muutenkin tavallista työläämpää: Kuitti pitää lähettää (ainakin Android-versiossa kuvana, pdf ei kelvannut) sekä ilmoittaa paino, jonka Posti kyllä tietää jo, koska se kerrotaan Postin omassa ilmoituskirjeessä. Web-tullaus tulli.fi:ssä ei näitä kysele, ja siellä voi tullata myös yli 150 euron lähetykset. Luonnollisesti vaikeimmille tapauksille eli täysin ilman seurantaa saapuville kirjeille tällaista edullisempaa mobiilivaihtoehtoa ei tarjota muutenkaan. Valitettavasti juuri kirjeiden kohdalla 2,90 € käsittelymaksu ja tullauksen lisävaiva tuntuvat kaikkein kohtuuttomimmilta suhteessa lähetyksen yleensä vähäiseen arvoon ja kokoon.[1]

Grillipostijonossa

Edellisessä kirjoituksessani ja palautteessa nousi esiin myös toinen pakettikaaoksen aiheuttaja: jakeluverkoston uudistuminen tai pitäisikö sanoa alasajo. Tämän seuraukset ovat vain pahentuneet tullausuudistuksen jälkeen.

[1] *Käsittelymaksu veloitetaan postilähetyksistä, joista tulli veloittaa veroja tai tullia tullausvaiheessa. Lähetyksen arvo lasketaan sen kuljetuskustannukset sisältävästä arvosta, ei vaikkapa myyntihinnasta, joten lähetyksillä on yleensä verotettavaa arvoa. Poikkeus sääntöön ovat tietyt yksityishenkilöiden toisilleen kohdistamat ei-toistuvat lahjat, jotka eivät ylitä 45 euron arvoa tai muita rajoituksia – niitä tulli ei verota, joten käsittelymaksuakaan ei makseta.*

Kuva: Fabrizio Bensch / Reuters

Oikeakätisten IBM ThinkPad TransNote CeBIT 2001 -messuilla Hannoverissa – näyttö käännettynä yleisöön.

– mutta rajoitus tuskin ainakaan auttoi kalliin yrityskoneen ostopäätöksissä. Nähdäkseni suurin ongelma oli kuitenkin vuonna 2002 sama kuin vuonna 2022. Suurelle yleisölle ajatus käsin kirjoitettujen digimuistiinpanojen tekemisestä oli vieras silloin ja vaikuttaa olevan lähes yhtä vieras nykyisin.

ISO 216 A5

Vaikka koronapandemia on karsinut tapaamisia, ihmiset kohtaavat edelleen myös kasvotusten. Muistelen kirjoitushetkellä edeltävän viikon fyysistä kokoustani, johon osallistui puolen tusinaa henkilöä. Ellei pakollisia maskeja ja koronapikatestejä lasketa, asetelma oli tuttu: Neuvotteluhuoneen pöydällä oli pari kannettavaa tietokonetta esityksiä varten, ja älylaitteisiin tehtiin kalenterimerkintöjä. Muistiinpanot sauhusivat kuitenkin paperisiin muistikirjoihin – lähinnä A5-kokoisiin, kovakantisiin. Nyt kun mietin, en muista työkokouksissa nähneeni kenenkään muun kuin itseni kirjoittavan käsialaa sähköistetylle alustalle. Näppäimistöjä toki käytetään joskus pöytäkirjanpitoon.

Paperin käytännöllisyys muistiinpano- ja hahmotteluvälineenä on kiistatonta. Muistikirjat ja kynät ovat kompakteja ja edullisia sekä vapaamuotoisissa tilanteissa sähkölaitteita ergonomisempia. Ne eivät tarvitse töpseliä, latausta, käynnistelyä tai välttämättä edes pöytää. Paperille kirjoittaminen ei myöskään juuri vie huomiota muilta asioilta. Sen sijaan paperimuistiinpanojen kattava digitalisointi ei ole oikein osoittautunut tarpeelliseksi, vaikka aikanaan yritinkin maalailla uhkakuvia hukkuvista paperinpaloista. Säästettäväksi tarkoitetut asiat saadaan kyllä talteen paperilta muutenkin, ja lopuilla ei ole niin väliksi.

Kokemukseni mukaan yhä useampi postilähetys päätyy nimittäin noutopisteeseen, sen sijaan että ne jaettaisiin perille, vaikka aiemmin vastaavat lähetykset tulivat suoraan perille. Lisäksi postin noutopisteverkosto koostuu nykyisin tilavien postikonttoreiden sijaan sekalaisesta seurakunnasta automaatteja ja asiasmiesposteja. Nykypostit löytyvät grillikioskeilta sekä parturi-kampaamoista. Ennen nopeasti postilaatikkoon tullut kirje (kyllä, myös kirje) saattaa päätyä nyt jollekin satunnaiselle huoltoasemalle toiselle puolelle kaupunkia.

Aiheesta syntyi talvella eläväistä keskustelua Skrollin lukijakanavilla (skrolli.fi/pikaviestimet). Monien lähetykset päätyvät jatkuvasti vaihteleviin noutopaikkoihin, koska yllättäen grillikioskissa on rajatusti tilaa. Käydyn keskustelun perusteella haarukoimme, että tiiviimmin asuttujen kaupunkikeskustojen alueella paketti voi päätyä uuteen paikkaan 2–3 kilometrin säteellä, Espoon tai Vantaan kaltaisessa autokaupungissa lisälenkkiä tullee 5–10 kilometriä ja maaseudulla jakelupisteen odottamaton muutos voi tarkoittaa matkaa yli 50 kilometrin päähän. Itsekin olen päässyt tekemään suoranaista kaupunkimatkailua postini perässä paikkoihin, joissa en ole koskaan aikaisemmin käynyt.

Useampi kommentoi myös, ettei koskaan tilaa lähetyksiä Postin pakettiautomaatteihin, koska automaattien täyttöasteen vuoksi lähetykset päätyvät minne sattuvat. Toisaalta lähetyksiä tuntuu päätyvän automaatteihin, vaikka niitä ei sinne tilaisikaan ja kiusallisen usein vieläpä niin, ettei avauskoodia ole mahdollista saada, koska lähetykseen ei kuulunut puhelinnumeroa tai sähköpostiosoitetta. Koodia ei saa myöskään asiakaspalvelusta. Rakentavana ideana nimimerkki **Razum** toi esiin vaihtoehtoisen pakettiautomaatin tai toimituspaikan valintamahdollisuuden – jos tavanomainen paikka on täynnä, olisi hienoa voida kertoa Postille itselle sopivin kakkosvaihtoehto.

Parannusta odotellessa Siperia opettaa oman alueen niksit. **Jupp3** totesikin kuvaavasti, että "monesti kannattaa tilata 'siihen toiseksi lähimpään' postiin."

IOSS-arpapeliä

EU-alueen ulkopuolelta tulevien lähetysten tullausta helpottamaan luotu IOSS-järjestelmä (*Import One-Stop Shop*) on myös osoittautunut melkoiseksi arpakoneeksi. Teoriassa sen pitäisi toimia niin, että lähettäjä maksaa kohdemaan ALV:n IOSS-järjestelmän kautta ja tämän jälkeen kohdemaan posti hoitaa (maksutta) tullauksen ja lähetyksen perille vastaanottajalle. Ainakaan omalla kohdallani näin ei tapahdu juuri koskaan, vaan joudun tullaamaan itse lähetyksen, josta kerrotaan tullauskirjeessä vain lähettäjän nimi, vastaanottopäivämäärä ja paino. Vastaanottaja ei voi tarkistaa itse pakettia, sen sisältöä tai osoitemerkintöjä mistään. Seurantakoodi annetaan myös, mutta harvoinpa kirjeiden lähettäjä sellaista ilmoittaa tai edes tietää.

IOSS:n olemassaolo itseasiassa li- ▶

Tällaiset havainnot eivät kuitenkaan ole lannistaneet sähköistettyjen muistiinpanohärveleiden keksijöitä. Erilaisia vaihtoehtoja on ilmestynyt läpi vuosien, ja jokuhan niitä on myös käyttänyt. Tämän Skrollin sivuilla 42–47 **Mitol Meerna** tarkastelee ohjelmapohjaisten muistivihkojen nykytilannetta. Itse puolestani seuraan tällä palstalla TransNoten jalanjäljissä sähköisen paperin kehityspolkuja. Valitsemani kuvakulmat muodostavat uskoakseni edustavan läpileikkauksen eri teknologioista ja tavoista, joilla paperia on viime aikoina pyritty digitalisoimaan. Havainnollistan aihetta keskeisten tuote-esimerkkien kautta, mutta vastaavia laitteita on toki paljon muitakin.

Moleskine Smart

Jokavuotinen työrituaalini on uuden Moleskine-muistikirjan ostaminen. Italialaisen Modo & Modon muistikirja näyttää niin ajattomalta, että yhtiön historian voisi kuvitella ulottuvan sadan vuoden taakse. Tosiasiassa Moleskine ei ole paljoa TransNotea vanhempi – ensimmäiset kirjat julkaistiin vuonna 1997. Sittemmin tästä pyöreäkulmaisesta, yleensä A5-kokoisesta muistikirjasta ja sen klooneista vaikuttaa tulleen lähes yhtä tyypillinen näky suomalaisilla palaveripöydillä kuin sinivalkoiset konttorikirjat olivat aikaisemmin.

Moleskinen muistikirjoja saa monenlaisilla sivumerkinnöillä: tyhjinä, pistekuviolla, ruudullisina, viivoitettuina ja erilaisina kalentereina. Tämän lisäksi ainoat "toiminnallisuudet" ovat kirjojen taakse piilotettu tasku sekä kuminauha, joka pitää kannen suljettuna. Moleskine myy myös kuminauhaan ruuvattavia metallisia kohokirjaimia, joilla kirjaa voi personoida – nimikirjaimeni näkyvät artikkelin aloituskuvassa. Vuonna 2012 yhtiö alkoi kuitenkin kosiskelemaan myös sähköisten mustiinpanojen markkinoita Evernote Smart Notebook -yhteistyötuotteella. Tämä tuote perustui samannimiseen pilvipalveluun ja sen valokuvia ottavaan mobiilisovellukseen.

Vuonna 2016 Moleskine lanseerasi oman Moleskine Smart -älykirjansa mobiilisovelluksineen (Android/iOS). Smartin ytimessä oli kamerapuhelimen sijaan Moleskine Pen+ -älykynä, jolla voi kirjoittaa paperille. Kynä synkronoitiin älylaitteeseen, mutta se toimi hetkellisesti myös irrallaan. Moleskinen paperiset "älykirjat", joita oli tavallisten muistikirjojen tapaan useita erilaisia, olivat vain muunnelmia yhtiön tavallisista muistikirjoista. Niiden sivuille painettiin arkin ja sivun tunnistekuvio sekä sähköpostin lähetyskuvake, jotka Smartin kynässä oleva kamera tunnistaa. Lisäksi kirjojen paperileikkausta pullistettiin ulkolaidoilta, jotta ne muistuttaisivat "paperitablettia".

Paria vuotta myöhemmin Moleskine päivitti järjestelmää Pen+ Ellipse -kynällä, ja tämä versio Smartista on myynnissä edelleen (moleskine.com/smart, hinta alkaen 249 €). Järjestelmään lisättiin myös Adobe Illustrator -integraatio, jonka myötä kynän voi kytkeä myös suoraan Windows- tai Mac-tietokoneeseen. Illustratoria varten julkaistiin jopa neliön muotoinen Creative Cloud Connected Paper Tablet -kirja, jossa oli erityisiä Illustrator-painikekuvia – se ilmestyi vuonna 2019 ja sinnitteli markkinoilla jotakuinkin yhtä kauan kuin IBM TransNote. Sittemmin erikoiskirja kuopattiin kaikessa hiljaisuudessa, mutta Illustrator-integraatio toimii muillakin yhtiön älykirjoilla. Älypuhelinpuolellakin Smart tallentaa kuvat vektorimuodossa ja taltioi automaattisesti jopa intervallivideot (*time lapse*).

Moleskine Smart on käytännössä

sää jopa kerroksen lisätyötä tällaisiin ilmoituksiin. Joskus tullausjärjestelmä nimittäin pyytää ilmoittamaan "Myyjän IOSS-tunnisteen" (*IOSS number*, tunnetaan myös etuliitteestä IM), vaikka ALV on maksettu etukäteen. Maksetun ALV:n määrää tai myyjän nimeä ei kuitenkaan kerrota – ainoa tunnistetieto on lähettäjän nimi, joka on jonkinlainen satunnaiselta vaikuttava nosto lähetyksen osoitetarrasta ja saattaa sisältää kokemukseni mukaan esimerkiksi pelkän pilkun (ks. Sirénin sisäpiiri Skrolli 2021.4), henkilön nimen tai vaikkapa yhdistelmän lähettäjän kadunnimestä ja sukunimestä. Jotkut myyjät ovat kuulemma alkaneet kirjoittamaan IOSS-numeroaan lähettäjän nimen kohdalle, jotta yksilöivä tieto tulisi perille vastaanottajalle lähettäjän nimen mukana, mutta kirjeiden osalta sellaisenkin läpipääsy lienee arpapeliä. Ainoaksi vaihtoehdoksi jää usein arvata, keneltä lähetys on tulossa, ja kysellä IOSS-tunnistetta häneltä.

Lähetyksen tapahtumat

Lähetyksen vastaanottajaa on tavoiteltu kirjeitse	28.1.2022 klo 18:26	NTF Notification Tool
Lähetyksen vastaanottajan puhelinnumeroa ei ole saatu	28.1.2022 klo 18:18	NTF Notification Tool
Muista tehdä tullaus ja maksaa Postin käsittelymaksu, jotta lähetys vapautuu Postin varastosta.	28.1.2022 klo 18:17	Postin varasto
Lähetys on rekisteröity.	19.1.2022 klo 17:42	Tietojen tallennus
Lähetys on lajiteltu.	31.12.2021 klo 12:19	VANTAA 01530
Lähetys on saapunut kohdemaahan.	31.12.2021 klo 12:19	VANTAA 01530
Lähetys on rekisteröity.	28.12.2021 klo 07:06	Tietojen tallennus
Lähetyksen ALV on maksettu ostovaiheessa. Posti hoitaa tulli-ilmoituksen puolestasi.	17.12.2021 klo 02:25	

Tyypillinen tilanne: Postin seurannan tullausohjeistus on pielessä. Toisin kuin Posti sanoo, se ei tehnyt – eikä ainakaan minun lähetysteni kohdalla juuri koskaan tee – tulli-ilmoitusta asiakkaan puolesta. Lähetyksestä ei myöskään pidä "muistaa maksaa" Postin käsittelymaksua, koska sitä ei makseta ALV-ennakkomaksetuista lähetyksistä. Postin käyttöliittymä tosin pyysi ystävällisesti maksamaan sen silti, jos käsittelymaksun tilannetta erehtyi tarkistamaan Postin tarkistussivulta. Kuten päivämääristä näkyy, aikaa palaa lisäksi kuukauden päivät kirjeen saapumisesta Suomeen, ennen kuin sitä voi edes yrittää tullata. Ei onnistu, vaikka yrittäisi. Ennen tällainen kirje kulki lähettäjältä vastaanottajalle viikossa.

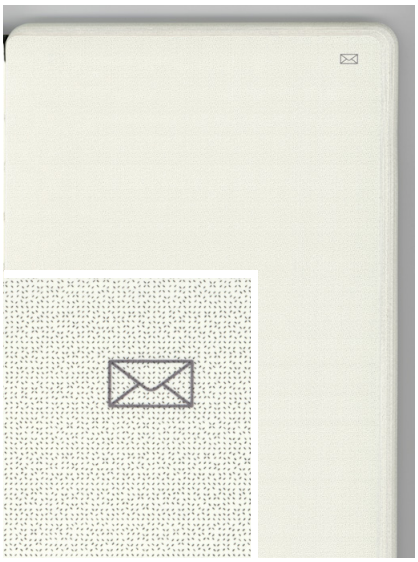
Myös IOSS-lähetysten arvon ilmoittamisessa tuntuu esiintyvän vaihtelua. Minulla lukuisien IOSS-lähetysten tullauksen kohdalla on vaadittu ilmoittamaan myös lähetyksen arvo. Yhdessä palautteessa kuitenkin kommentoitiin, että itse tullattavaksi jääneessä IOSS-lähetyksessä ei ollut tarvinnut ilmoittaa summaa. Liekö eroa siinä, onko lähetys kirjepostia tai mahdollisesti mistä se on tulossa. Minulta arvoa on pyydetty ilmoittamaan etenkin Royal Mailin kirje- tai maksikirjepostille. Pikanttina lisänä Iso-Britannian puntaa ei ole tulli.fi:n yleisimpien valuuttojen listassa.

Asiat voisivat kai olla huonommin kin. Huonoja uutisia on nimittäin alkanut kantautua muualtakin kuin meiltä. Retroyhteisöissä on nostettu esiin muun muassa Ruotsin paikallinen postilaitos, joka on veloittanut esimerkiksi

moderni tulkinta IBM:n TransNotesta, jossa digitointi on siirretty alustasta kynään ja tietokoneen korvaa älypuhelin. Sitä mukaa kun älykynällä kirjoitetaan paperille, kynä seuraa liikettä sekä siirtää tekstin ja piirrokset mustavalkokuvana älypuhelimeen tai tietokoneelle, josta ne voidaan jakaa edelleen esimerkiksi pilveen. Koska paperina käytetään erityistä Moleskine-kirjaa, Smart tunnistaa automaattisesti myös sivunumerot ja osaa näin erotella älypuhelimen päässä eri sivuille tehdyt muistiinpanot toisistaan – paikaten näin yhden TransNoten puutteista. Moleskine suorittaa käsin kirjoitetulle tekstille myös tekstintunnistuksen ja osaa tallentaa ääntä muistiinpanojen rinnalla.

Periaatteessa paperi voi olla muutakin kuin Moleskinen tarjoamaa, mutta siihen täytyy olla painettuna tai tulostettuna sopiva Ncode-taustakuviointi. Ilman tätä pohjatekstuuria älykynä ei toimi lainkaan. Moleskine käyttää kynässään Neo Smartpenin tekniikkaa, joten kuvioon voi tutustua tarkemmin Neon vapaasti ladattavissa pdf-dokumenttipohjissa (neosmartpen.com/en/ncode-pdf). Jotkut väittävät myös saaneensa Neon paperit toimimaan Moleskinen mobiilisovelluksen kanssa. Itse en ole siinä onnistunut – Moleskine on viime vuosina saattanut rajoittaa sovellusta – mutta Neon itsetulostettavat paperit toimivat minullakin hyvin Moleskinen tietokoneintegraation kanssa.

Moleskine Smart on toimiva laajennus perusmuistikirjaan, jos voi elää sen kanssa, että Pen+ Ellipsen micro-USB-liittimestä ladattava akku kestää vain 4–5 tuntia – eivätkä ne 20–30 € maksavat "älykirjatkaan" ilmaisia ole. Mikäli Moleskine Smart kiinnostaa, kannattaa tutustua myös tuotteen taustalla olevaan Neo Smartpen -perheeseen (neosmartpen.com). Neo on hieman teknisempi vaihtoehto Moleskinen trendipaketille.



Kuva: Janne Sirén

Moleskine Smartin Ncode-koodausta.

Iskn Repaper

Moleskinen älykkäästä muistiinpanojärjestelmästä on teoriassa jopa piirtoalustaksi, mutta sen vektorikuvien laatu ei välttämättä riitä graafikoille. Moleskinen Illustrator-integraatiosta uupuu paineentunnistus, ja vaikka mobiilisovellus tunnistaa paine-erot, sovelluksen vektoritallenteet saattavat olla turhan likaisia kuvatuotantoon. Piirtoalustaksi on siis parempiakin vaihtoehtoja. Yksi tällainen tulee ranskalaiselta Iskn-yhtiöltä: Iskn Repaper (iskn.co, alkaen 199 €) on hieman A5-sivua suurempi digitointialusta, joka kytketään PC-/Mac-tietokoneeseen tai Android/iOS-älylaitteeseen. (Laitteesta on lisäksi edullisempi 129 euron versio, joka toimii vain tietokoneeseen kytkettynä.) Tuoteperhe tunnettiin aikaisemmin Slate-nimellä.

Repaperille asetetaan paperiarkki piirtämistä varten. Laitteen mukana tulee lehtiö, mutta paperi voi olla muutakin. Laite toimii myös minkä

Amiga Addict -lehdestä ALV:n kahdesti – sekä IOSS-lähetystä lähetettäessä että vastaanottajalta. Irlannissa Fusion Retro Booksin IOSS-lähetyksistä on myös veloitettu sekä tupla-ALV että peräti 3,50 € käsittelymaksu. Tällaista ei sentään Suomesta ole kuulunut. Yritän silti hillitä orastavaa kiitollisuuttani tullipostia kohtaan, koska epäilen Tukholman syndrooma -pikatestin näyttävän positiivista.

Kukapa olisi globalisaation ja digitalisaation edetessä uskonut, että kaikki tämä kehitys johtaa siihen, että vuonna 2022 tullaamme paperikirjeitä? Yksistään se työmäärä, joka yksittäisen kirjeen vastaanottamiseen voi nykyisin liittyä, on saanut minut harkitsemaan vakavasti henkilökohtaisen digiloikkani kiihdyttämistä turbo boost -vaiheeseen.

Entä se GDPR ja hukattu kirjeposti? GDPR koskee yhtäläisesti kaikkea dataa – digitaalista ja paperista. Toimijoilla on siis yhtäläinen velvollisuus ilmoittaa julkisesti myös paperisista aukkokohdista, minkä sekä Kela että Posti edellä mainitussa tapauksessa tekivätkin.

Selviytymisopas postikaaokseen

Apua! Minulla ei ole EU-alueen ulkopuolelta tulevalle lähetykselle seurantanumeroa eikä tullauskirjettä?

Kokemukseni mukaan tullauskirjeen saapuminen voi kestää kuukaudenkin lähetyksen saapumisesta Suomeen, joten se voi olla vielä matkalla. Tämän jälkeen on 20 päivää aikaa suorittaa tullaus, joskus hieman enemmän. Koska kirjeitä lähetetään vain yksi, sen hukkuminen on kuitenkin mahdollista. Tarvittaessa seurantanumeroa kannattaa kysellä lähettäjältä, sillä sen avulla voidaan sekä aikaistaa tullausta jonkin verran että välttää lähetyksen palautuminen tullauskirjeen kadotessa. Jotta tietää keneltä kysellä ja milloin, lähetyksistään kannattaa pitää entistä tarkemmin kirjaa – ei kannata vain odotella saapumisilmoituksia.

Myös joidenkin ulkomaiden postien (esim. Royal Mail) lähetyksiinsä liittämä tunnistenumero voi auttaa, koska Suomessa tätä numeroa käytetään usein tullauksen seurantaan, vaikka lähettäjämaan posti ei lähetykselle seurantaa tarjoaisikaan. Parasta on, jos lähettäjällä on mahdollisuus esittää kuva osoitetarrasta ja siinä mahdollisesti olevasta viivakoodista. Todennäköisesti (mutta ei aina) tässä viivakoodissa on se koodi, jolla lähetyksen seuranta ja tullaus Suomessa tapahtuu. Olen itse pelastanut pari lähetystä saatuani lähettäjän kaivelemaan lähetyspostinsa kuittiarkistoa.

Jos käytät OmaPosti-sovellusta (posti.fi/fi/henkiloille/omaposti), kannattaa tarkistaa myös siellä näkyvät tiedot. Joskus kansainvälisestä lähetyksestä voi tihkua tietoja myös sovellukseen saakka. Kenties jopa edullisempi tullausmahdollisuus alle 150 euron lähetyksille.

Minulla on seurantanumero, mutta ei Postin kirjettä tullauksessa tarvittavine tietoineen?

Postin pakettiseuranta (posti.fi) ▶

tahansa sopivan kokoisen kynän kanssa, kynän varteen kiinnitettävän Repaper Ring -lisäosan (39 €) avulla – tuotteesta onkin kynävalmistaja Faber-Castellin kanssa tuotettu yhteistyöpainos. Monet kuulakärkikynät alkavat kuitenkin olla liian paksuja. Vaihtoehtoisesti voi käyttää Repaperin omia kyniä, jotka eivät edellytä lisäosia. Repaperille voi piirtää myös osoitinkynällä ilman paperia, jolloin se toimii perinteisenä tietokoneen piirtopöytänä. Kuten TransNote tai Moleskine Smart, Repaper toimii tarvittaessa myös ilman emolaitetta. Sen micro-USB-liitimestä ladattava akku kestää kuutisen tuntia ja muistiin mahtuu kymmenisen kuvaa (microSD-muistikortin kanssa enemmänkin).

Kuva: Janne Sirén

Iskn Repaper sekä erilaisia Repaper Ring -kyniä keskellä. Laidoilla täysdigitaalisia vaihtoehtoja: Apple Pencil ja iPad Pro -tabletti (vas.) sekä Galazy Z Fold3 -älypuhelin S Pen Fold Edition -kotelon kera. Näistä vain Applen kynä vaatii latausta – sen perässä on korkin alla latausliitin. Uusimmat iPadit tukevat myös langattomasti latautuvaa Apple Pencil 2:sta, joka kiinnittyy tabletin laitaan. Myös S Pen tarvitsee virtaa, mutta se saa virtansa langattomasti älylaitteen näytöltä käytön aikana. Repaperin kynissä on pelkkä magneetti.

Testissäni Repaper ei tuottanut yllätyksiä: Sekä kynällä että osoitinkynällä tehdyt vedot siirtyivät uskollisesti ja paineen huomioiden tietokoneelle. Samoin tavalliseen kynään pujotettu Repaper Ring tunnisti vedot ja paine-erot samalla pieteetillä. Alustan painikkeilla kuva voidaan myös jakaa viidelle eri tasolle (*layer*) alustan ollessa ir-

kertoo osan vastaavista tiedoista. Se saattaa esimerkiksi kertoa paketin kokotietoja, mikä voi olla arvokasta lähetyksen tunnistamiseen. Palvelu ei kuitenkaan kerro viimeistä noutopäivää tai lähettäjän nimitietoa, jotka ovat Postin ilmoituskirjeessä tai sen sähköisessä vastineessa. Posti lähettää tällaisia fyysisiä kirjeitä vain yhden, joten jos se on viivästynyt tai hukkunut, eikä sähköistä ilmoitusta ole tullut, tämä voi olla aito ongelma. Itselläni pari postilähetystä on palannut lähettäjälle, koska en koskaan kuullut niistä.

Seurantanumero pelastaa kuitenkin paljon. Postin käsittelymaksun maksupalvelu (posti.fi/fi/henkiloille/paketit-ja-seuranta/paketin-tullaus-ja-kasittelymaksu) kertoo lähetystä seurantanumerolla haettaessa painon, saapumispäivän ja viimeisen noutopäivän – tiedot näkee ennen kuin tarvitsee maksaa mitään. Tullin palvelu (asiointi.tulli.fi/asiointipalvelu/import) puolestaan kertoo lähettäjän nimitiedon. Näin saat käyttämällä "kätevästi" kolmea eri verkkopalvelua kaikki samat tiedot kuin kirjeessä. Enempää tietoa et sitten voikaan saada mistään, paitsi lähettäjältä, joten onnea lähetyksen sisällön arvailuun.

Minulla on seurantanumero, mutta Postin tai Tullin palvelut eivät tunnista sitä?

On mahdollista, että seuranta ei ole vielä rekisteröitynyt tullattavaksi, koska tämä tapahtuu viiveellä. Tällöin ei auta kuin yrittää myöhemmin uudelleen. Postin pakettiseurannan pitäisi kuitenkin tunnistaa koodi jo varhain. Jos näin ei käy, on mahdollista, että seurantanumero on vaihtunut matkalla esimerkiksi monesta eri toimijasta koostuvan postitusketjun kohdalla.

Parcelsin (parcelsapp.com/en) kaltaiset useita seurantajärjestelmiä seuraavat palvelut osaavat usein kertoa vanhalla seurantanumerolla, mikä on uusi seurantanumero. Voit sitten kokeilla edellä mainittuja neuvoja tällä uudella numerolla. Kannattaa kuitenkin tiedostaa, että kolmannen osapuolen palvelut voivat kerätä seurantatietoa, joskaan itse en ole mitään ilmeistä harmia huomannut. Välttäisin kuitenkin tarpeetonta käyttöä.

Pakettini ulkomailta päätyi pakettiautomaattiin, vaikka lähettäjällä ei ollut puhelinnumeroani tai sähköpostiani?

Tervetuloa kerhoon. Et voi saada automaatin noutokoodia mitenkään, ellei sitten lähettäjä ole merkinnyt omia yhteystietojaan vastaanottajan tietoihin ja toimita koodia sinulle. Yleensä näin ei ole, vaan todennäköisesti Postin järjestelmä on vain keksinyt sinulle jonkun sähköpostiosoitteen tapaisen ja luulee, että sinut tavoittaa sieltä. Tätä tapahtuu. Olen kerran onnistunut maanittelemaan Postin asiakaspalvelun kertomaan osoitteen, jonka postijärjestelmä oli lähetykseeni liittänyt ja jonne noutokoodini oli lähetetty. Se edusti tyylisuuntaa älä-vastaa@posti.com.

Ainoa, mitä voit tehdä, on soittaa Postin asiakaspalveluun ja pyytää siirtämään paketti automaatista lähimpään asiamiespostiin. Varaudu tosin siihen, että paketti päätyy takaisin automaattiin, koska järjestelmässä on edelleen se leikkisähköpostiosoitteesi. Seuraa seurantaa ja toista ohje tarvittaessa. Älä suostu asiakaspalvelijan ehdotukseen pyytää lähettäjää ilmoittamaan jälkikäteen yhteystietosi Postille. Kokemukseni mukaan ulkomaisen lähettäjän Suomeen tekemä ilmoitus hylätään. Ulkomaan postin tapauksessa tietoja voi ilmoittaa vain lähettävä postiorganisaatio, ja sitä kautta tehdyssä ilmoituksessa kestänee niin kauan, että paketti lähtee paluumatkalle.

rallaan, joskaan Moleskinen tapaiseen muistiinpanojen sähköistämiseen Repaperista ei ole kuin satunnaisesti. Sivunvaihto vaatii sekä paperin vaihtamista että painikkeiden painelua, eikä laitteen ohjelmistoakaan ole valjastettu tarkoitukseen. Repaper soveltuu tietokonetaiteilijalle, jonka tarkoituksena on piirtää luonnoksia tai ääriviivoja paperilla. Vektorikuvan lisäksi Repaper taltioi myös intervallivideon työn vaiheista – sitä ei skannerilta saa.

Iskn:n kilpailijoita ovatkin Moleskinen sijaan esimerkiksi Apple (iPad ja sen Apple Pencil -osoitinkynä) sekä tietenkin perinteikäs Wacom (wacom.com), jonka nimi on ollut 1990-luvulta lähtien synonyymi tietokoneeseen liitetylle piirtopöydälle. Wacom myy edelleen Intuos-sarjan piirtoalustoja ja myös malleja, joihin voi liittää paperin. Wacomin panostussuunta tuntuu kuitenkin olevan näytöllisissä piirtotableteissa, joiden hinta nousee 400–500 eurosta tuhansiin. Yhtiön teknologiaa löytyy lisäksi muun muassa Samsung Galaxy -älypuhelinperheen S Pen -kynän takaa.

E Ink

Matkalla paperittomaan maailmaan tasan kaksi hetkeä ovat tuntuneet itsestäni taikuudelta: Ensimmäinen oli Etch A Sketch -piirtolelun saaminen lapsena. Toinen oli Amazon Kindle 2 -lukulaitteen purkaminen paketista vuonna 2009 ja kuva sen sammutetulla sähkömustenäytöllä. Tämä ei ole sattumaa, sillä laitteet perustuvat etäisesti toisiaan muistuttavaan tekniikkaan.

Toisin kuin yleisesti uskotaan ja jopa mainostetaan, Etch A Sketch ei perustu magneetteihin. Sen "näyttönä" toimii lasilevy, jonka sisäpinta on peitetty vaalealla alumiinijauholla. Käyttäjä voi ohjainrullien liikutteleman plotterin avulla raaputtaa tätä jauhoa, jolloin se tippuu alemmaksi ja seurauksena on tummia raitoja. Alumiinijauhon saa nostettua takaisin paikoilleen ravistelemalla laitetta. Kindlen näyttö taas nostelee läpinäkyvälle pinnalleen vuorollaan vaaleita tai tummia hiukkasia sähköisesti.

Alkuperäinen sähkömustenäyttö kehitettiin edellä mainitun George E. Paken johtamassa Xeroxin PARC-tutkimuskeskuksessa jo 1970-luvulla. Samassa paikassa kehitettiin samoihin aikoihin myös moderni graafinen käyttöliittymä ja tekstinkäsittely (osin toki Stanford Research Instituten 1960-luvun *The Mother of All Demos* -työn pohjalta). Useimpien PARCin keksintöjen laajempi kaupallistaminen jäi kuitenkin muille, vaikka Xerox julkistikin joitakin (liian) kalliita tuotteita. Sähkömusteen kohdalla jouduttaisiin odottamaan peräti vuoteen 2007, jolloin E Ink -yhtiö esitteli oman tulkintansa ideasta.

E Inkin sähkömustenäyttöä käytetään nykyisin Amazon Kindlessä ja lukuisissa muissa laitteissa. Kilpaileviakin sähkömusteita on sittemmin tullut ja mennyt, mutta ei juuri jäänyt. E-ink-näyttö on täytetty mikrokapseleihin pakatuilla tummilla ja vaaleilla hiukkasilla, joita saadaan nostettua ja laskettua sähkövarausta muuttamalla. Mikrokapselit ovat pyöreitä, joten kuvasta ei erotu pikselirasteria, vaan korkeintaan (muste)pisaroita; tarvitaan suurennuslasia, että kennomainen kapselirakenne näkyy himmeästi. Näyttö on bistabiilinen eli hiukkaset pysyvät itsestään kummassa tahansa tilassa. Sähköä tarvitaan vain tilan muuttamiseen, mikä tekee näytöstä energiatehokkaan – ja taikuutta.

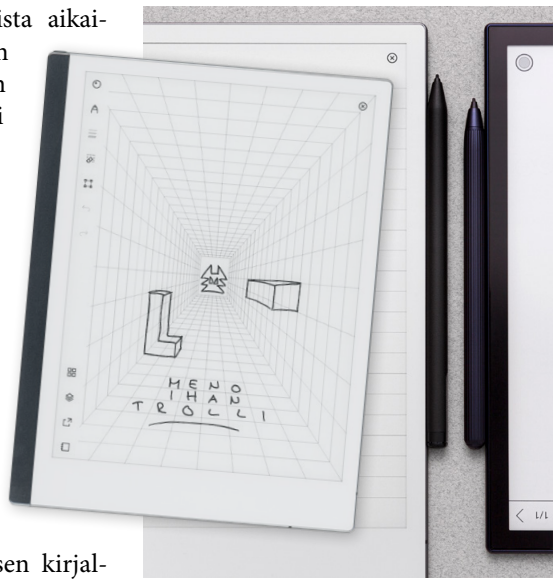
Koska Etch A Sketchissä tai e-ink-näytössä ei ole taustavaloa, niiden katselu ei myöskään rasita silmiä perinteisten näyttöjen tapaan. Sähkömustenäytön kuva erottuu lisäksi tavallisia näyttöjä paremmin kirkkaassa auringonpaisteessa. Miinuspuolena sähkömustenäytön päivittäminen on verrattain hidasta sekä altista aikaisemmin näytettyjen kuvien jättämille haamuille, joskin vain väliaikaisesti. Lisäksi sähkömustenäytöt ovat pääasiassa harmaasävynäyttöjä, tosin poikkeuksiakin on – lisää e-ink-näyttöjen väreistä artikkelin lopun laatikossa.

Olen ollut Kindlen tyytyväinen käyttäjä läpi kymmenen rautasukupolven. Amazon on hionut sähköisten kirjojen hankinta- ja lukuprosessin niin saumattomaksi, että ainakin englanninkielisen kirjallisuuden osalta olen koukussa. Amazonin lähestymistapa sähkömusteeseen on kuitenkin ollut viime vuodet konservatiivinen ja heiltä lienee turha odottaa dramaattisia lisäinnovaatioita tällä alueella. Laajempia sähkömusteen käyttömahdollisuuksia joutuu etsimään muualta.

Valaistu sähkömuste: Vaikka e-ink-näytöissä ei ole taustavaloa, osa laitteista sisältää valaistun näytön, esimerkiksi nyky-Kindlet sekä Boox Note Air2 ja Nova3 Color. Kyseessä on säädettävä ja valinnainen etuvalokalvo, jossa valo heijastetaan näytön päälle sivuilta. Se ei näin ollen loista suoraan silmiin. Nykyisin voidaan yleensä säätää sekä kirkkautta että värilämpöä. Etuvalollisissa sähkömustenäytöissä on kuitenkin paljon mallikohtaista ja yksilöllistäkin laatuvaihtelua, kuten kirjoitin kolumnissani Elämäni kuolleet pikselit (Skrolli 2016.2). Herkimmillä asiakkailla ne ovat johtaneet palautus- ja vaihtokierteisiin, toiset taas eivät huomaa eroa. Vaikka laatutaso on keskimäärin parantunut, palstan laitteista Boox Note Air2 -yksilön näytön valoisuus on selkeästi suurin toisessa alakulmassa. ReMarkable 2:n näyttö on turvallisin – siinä ei ole valoa lainkaan.

ReMarkable 2

Yksi sähkömusteinnovaattori on norjalaisen **Magnus Wanbergin** perustama reMarkable (remarkable.com), joka sai lähtölaukauksensa vuoden 2016 joukkorahoitus- tai oikeammin ennakkotilauskampanjasta. Toisen su-



Kuvat: Janne Sirén

ReMarkable 2:ssa on magneettinen stylus. Oikealla pilkistää samantapainen Boox Note Air2 -sähkömustetabletti Androidilla (s. 33). Kummankaan osoitinkynä ei vaadi lataamista.

kupolven tuote julkistettiin vuonna 2020, joskin pandemian vuoksi toimitukset ovat tasaantuneet tasaiseksi saatavuudeksi vasta viime aikoina. ReMarkable 2 (alkaen 399 € tai 299 € pilvitilauksen kanssa) on hieman A5-paperia suuremmalla 10,3 tuuman e-ink-näytöllä (1872×1404, 226 dpi) varustettu kosketusohjattu tabletti.

Tablettia ohjataan passiivisella kynällä, jonka painalluksen voimakkuuden laite kykenee erottamaan 4096:n vaihtoehdon tarkkuudella. Kynä ei vaadi latausta, eikä itse laitteenkaan akkua tarvitse jatkuvasti ajatella, sillä kestoa lupaillaan pari viikkoa – lataus tapahtuu USB-C-liittimellä. Käyttöjärjestelmänä on reMarkablen oma Linux-pohjainen Codex, jonka käyttöliittymä on tuunattu muistiinpanojen ottamiseen, organisoimiseen ja jakamiseen. Tietojen jakaminen pilveen onnistuu WLAN-yhteyden kautta, mutta melko suurena kustannusmiinuksena se vaatii lisäksi erillisen Connect-tilauksen (7,99 €/kk).

Ohjelmistoon kuuluu erittäin pätevä käsinkirjoitus- ja piirtotoiminto sekä mahdollisuus kerrostaa sivulle tekemiään muistiinpanoja ja raahata niitä uusiin paikkoihin. Tarjolla on myös runsaasti asiakirjapohjia, jotka muistuttavat Moleskine-muistikirjojen valikoimaa kirjakaupassa; reMarkableen voi esimerkiksi asentaa täysin käsikäyttöisen kalenterin, mikäli on sellaista tottunut käyttämään paperilla, ja jopa nuottipohja löytyy. ReMarkable toimii lisäksi e-lukijana, kunhan käytetään DRM-vapaita epub- tai pdf-tiedostoja – niihin voi lennosta myös lisätä tyhjiä sivuja muistiinpanoja varten. Tietokoneelle asennettavan selainlaajennuksen avulla laitteeseen voidaan lähettää myös web-artikkeleita luettavaksi. Toiseen suuntaan reMarkablen kuvan voi jakaa tietokoneelle vaikkapa presentaatiokäyttöön.

Itselleni suurin kysymysmerkki reMarkable 2:n kohdalla oli tekstinsyöttö käsin sekä sähkömusteen nopeus muistiinpanojen organisoinnissa. Uutiset ovat hyviä: reMarkable 2:lla kirjoittaminen ja piirtäminen oli testissäni todella sujuvaa, kuten myös asioiden valitseminen ja siirtely. Osoittaminen toimii hyvin käsialatarkkuudella, joskin aivan pienintä syheröä sillä on vaikea piirtää. Pinnaltaan paperimaiseksi karkaistu näyttö seuraa osoitinkynän liikettä nopeasti, eikä käyttöliittymä juurikaan kärsi sähkömusteelle tyypillisistä haamukuvista. Muistiinpanojen kirjaaminen ja siirtely on sujuvaa ja Connect-pilvitilaukseen kuuluva käsialan tunnistus toimi luotettavasti jopa suomeksi. Pilvitilaukseen kuuluu myös Google Drive -, Dropbox- ja OneDrive-integraatiot – oma käyttötapani onkin pallotella muistiinpanojani pilven kautta riippuen siitä missä niitä on helpoin kulloinkin käsitellä.

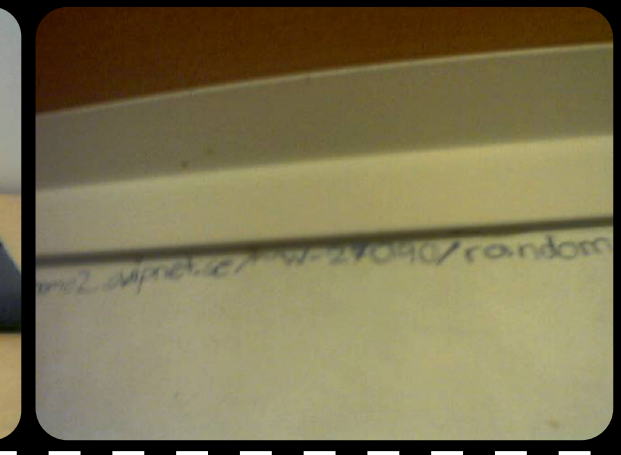
Kuten useissa muissakin osoitinkynissä, reMarkable 2:n kynän pään voi vaihtaa, ja tätä suositellaankin tekemään säännöllisesti paperimaisen kirjoitustuntuman takaamiseksi. Pakettiin kuuluu 9 vaihtopäätä. Laitteelle on saatavissa lisäksi Marker Plus -osoitinkynä (artikkelin kuvissa), jonka toisessa päässä on pyyhekumitoiminto. Tekstiä voi näin ollen kirjoittaa ja pyyhkiä lyijykynän tapaan, mikä on intuitiivista ja kätevää. Pyyhekumillinen osoitinkynä ei toki ole mikään uusi keksintö, mutta tässä joukossa harvinaisuus. Muilla palstalla mainituilla valmistajilla vastaava löytyy vain Booxin lisävarustevalikoimasta – edes Apple Pencilissä tai Samsungin nykykynissä kumia ei ole, vaikka painikeratkaisuita pyyhkimiseen löytyykin. Isot poistot on toki helpoin tehdä alueita pyyhkivillä toiminnoilla ja ne saa tehtyä peruskynälläkin. Vetokohtainen *undo* on myös todella kätevä.

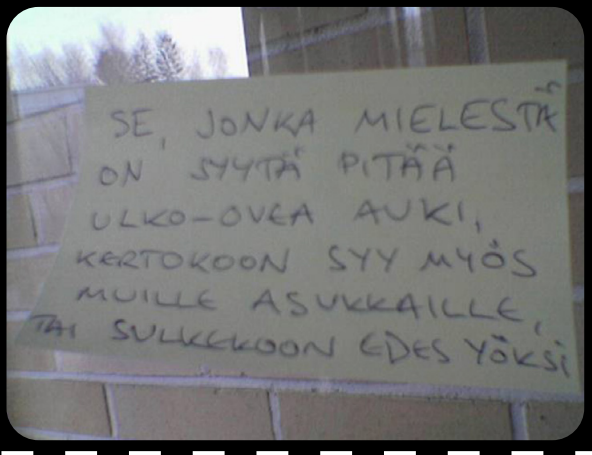
Nokia 7650 ja Z Fold3

IBM ThinkPad TransNoten ja sen seuraajien suurin haastaja ilmestyi kuitenkin jo puoli vuotta TransNoten jälkeen: Nokia 7650 (2001), ensimmäinen älypuhelin kameralla.[1] Ikuistin itsekin tällä puhelimella muistiinpanoja. Sittemmin lukuisat muut kameralliset älypuhelimet ovat ajaneet taskuskannerin virkaa. Arvelenkin, että kamerapuhelimesta on tullut merkittävin paperimuistiinpanojen digitalisoija. Vaikka kännykkään on vaivalloista naputella muistiinpanoja, asia kuin asia saadaan sähköistettyä sen kameralla.

Modernien älypuhelimien kamerasovelluksissa on jopa skannaussimulaatio ja tekstintunnistus. Kameroiden kaveriksi myydään myös erityisiä pyyhittäviä muistikirjoja, joita on mahdollista käyttää monta kertaa. Esimerkiksi Rocketbook Core (myrocketbook.eu, alkaen 38 €) voidaan pyyhkiä puhtaaksi märällä liinalla ja Rocketbook Wave

[1] *Nokia 7650 oli ensimmäinen kansainvälisesti merkittävä kamerapuhelin ja älypuhelin. Se ei kuitenkaan ollut maailman ensimmäinen kamerapuhelin. Itä-Aasian markkinoilla julkaistiin muutama kamerallinen peruspuhelin vuosina 1999–2000, joskin kamera saattoi olla esimerkiksi kohdistettu videopuheluita varten tai kuviin pääsi käsiksi vain PC-kaapelilla.*





◀ Tähän ei sähköpaperi pysty: Kamerapuhelin taltioi tekstejä myös ruotsinlaivan hytistä ja kerrostalon ulko-ovesta. Kameroina Nokia 7650 (2001, vasemmalla) ja Nokia 6600 (2003). Molempien VGA-tarkkuus (640×480) jätti toivomisen varaa, mutta jälkimmäisen laatu oli jo käyttökelpoista.

Kuvat: Janne Sirén

(33 €) tyhjenee mikroaaltouuniin laittamalla. Rocketbookilla on oma mobiilisovellus, joka tekee sivujen kameraskannauksesta ja arkistoimisesta varsin vaivatonta.

Pisteenä i:n päällä älypuhelimessa voi nykyisin olla kameran lisäksi sisäänrakennettu osoitinkynä näytölle kirjoittamista varten, kuten Samsungin vanhassa Galaxy Note -perheessä tai helmikuussa julkistetussa Galaxy S22 Ultrassa. Jos tavallisen älypuhelimen näyttö käy ahtaaksi, paineherkän kynäohjaimen saa myös taitettavalle Galaxy Z Fold3:lle (2021). Kokeilin palstaa varten myös tämän S Pen Fold Editionin toimintaa, enkä juuri löytynyt moitittavaa. Näytön taitekohta tekee toki laitteen pitelystä ja pitkistä vedoista tablettia hankalampia, eikä sileä näyttö ole yhtä paperimainen kuin reMarkablen karheampi sähköpaperipinta, mutta eivät nämä ylitsepääsemättömiä ongelmia ole.

Myös iPadin Apple Pencil on toiminut hyvin sekä omissani että etenkin muiden käsissä – Applen kynä on niittänyt mainetta graafikkopiireissä, ja sillä on piirretty tämänkin Skrollin grafiikoita. Koska useimmilla ihmisillä on jokin yleiskäyttöisempi älylaite aina mukana, herääkin kysymys, tarvitaanko erillisiä muistiinpanotuotteita mihinkään?

Kolmas laite

Kaikki testaamani paperin sähköistäjät suoriutuivat keskeisimmästä tehtävästään vallan mainiosti: Moleskine Smart on paperisen muistikirjan käyttäjän digiarkisto, Iskn Repaper on paperia tukeva digitaalinen piirtoalusta visuaalista työtä tekeville ja reMarkable 2 on esimerkki täysin sähköisestä paperilehtiön korvikkeesta. Pidän niistä kaikista. Kysymys on lähinnä siitä, kuinka suurelta osin ne kykenevät syrjäyttämään perinteistä paperinkäyttöä tai häviävätkö ne tässä yrityksessä jollekin yleiskäyttöisemmälle älylaitteelle.

Repaper on selkeästi pienen erityiskategorian tuote, mutta sekä Moleskine Smartissa että reMarkable 2:ssa on tuotetyyppiensä puolesta myös laajempaa arkipotentiaalia. Jos käyttää muutenkin muistikirjoja, sellainen on todella helppo korvata Moleskine Smartin tapaisella kirjalla ja älykynällä, kunhan muistaa ladata kynän säännöllisesti – nimittäin kun kynä ja vihko ovat mukana, mutta akku on tyhjä, tulee miettineeksi miksi vaivautua. Pääasiallinen motivaatio-ongelma onkin siinä, että suurimman osan ajasta paperinen muistikirja tavallisella kynällä riittäisi sellaisenaankin.

Osoitinkynällä varustetussa sähkömustetabletissa on kolmikosta eniten disruptiopotentiaalia. Tällaisen tabletin onnistunut ujuttaminen omaan työkulkuun voisi teoriassa murtaa kaikki paperin viimeiset linnakkeet – ne, joita tietokoneet tai älylaitteet eivät ole vielä kyenneet korvaamaan: muistikirjat, kirjat, lehdet, hitaammat hetket. Lcd- ja led-näytöiltä lukeminen ei ole kaikkein mukavinta ja toisaalta älypuhelinten näytöt ovat tarkoitukseen myös verrattain pieniä. Riittävän kookas mukana kulkeva sähkömustenäyttö voisi viimeinkin tarjota paperin veroista käytettävyyttä.

Vallankumouksen edessä on kuitenkin esteitä, vaikka laitteet muistaisi ladatakin. Esimerkiksi reMarkable 2 on muistiinpanokategorian lippulaivatuote, mutta sähkökirjojen ykköstarjoajan Amazon Kindlen kirjaston lukemien ei ole sillä mahdollista, paitsi vaivalloisesti kirjatiedostojen suojauksia purkamalla. Kindlestä taas ei ole muistiinpanojen ottajaksi, eikä kumpikaan laitteista oikein sovellu ammattikirjallisuuden käyttöön värinäytön puutteen ja niiden erilaisen jakelutavan vuoksi. Laajemmin yhteensopivia Google Play -sovelluskauppaa tukevia sähkömustetablettejakin on olemassa, mutta niiden kohdalla käytettävyys jää edellä

Kuva: Janne Sirén

Tapaus MBCD

Siirtymään paperilta digitaaliseen liittyy käyttötottumusten lisäksi tekniikkaa ja juridiikkaakin. Eräs pioneeri oli MikroBitti-lehti, joka julkaisi vuosina 1996–2001 vuosikertojensa 1994–2000 pdf-lehdet CD-ROM-levyllä. Kyseessä oli erikseen myytävä tuote, ei siis lehtitilaukseen kuulunut *Huvi- ja hyötyromppu* -ohjelmakokoelma. Romppulehteä kutsuttiin alkuun *Sähköiseksi lehdeksi* ja sitten *Virtuaalilehdeksi*; loppuhuipennus oli *(Parhaat) Bittivuodet 1994–2000* -kokoelma. Lyhyemmin MBCD1–MBCD6. Olin pienessä roolissa ensimmäisen, vuonna 1996 julkaistun tuplavuosikertalevyn *Sähköinen lehti 1994–1995* työryhmässä.

1990-luvun puolivälissä tekniikka alkoi viimein olla kypsää näköislehtien sähköiseen julkaisuun. MikroBitti oli siirtynyt paperitaitosta sähköiseen taittamiseen suomalaisittain etunenässä jo vuosikymmenen alussa, joten taitto-ohjelmasta saatiin nyt suoraan vuoden 1993 uutuuksia eli pdf-tiedostoja, joskin mainokset poistettiin. Ajankohta valikoitui ensimmäisen levyn alkupisteeksi myös markkinasyistä, olihan tilarikas CD-ROM-tekniikka tuon hetken kuuma juttu.

Aloitusajankohdalle oli kuulemani mukaan kuitenkin myös sopimustekninen selitys: vuoden 1994 alusta MikroBittiin kirjoittaneiden sopimuksia muutettiin huomioimaan ja sallimaan sähköinen uudelleenjulkaisu. Sittemmin vuonna 2013 voimaan astuneen tekijänoikeuslain pykälä "Arkistoidun ohjelman ja lehden uudelleen käyttäminen" sallii julkaisijalle ennen vuotta 1999 ilmestyneiden lehtien sähköisen uudelleenjulkaisun ilman tällaista erillislupaakin, minkä vuoksi tätä vanhemmatkin MikroBitit ovat nykyisin tarjolla lehden kotisivuilla skannatussa muodossa.

Vaikka CD:llä oli suhteellisen paljon tilaa, sekin loppui nopeasti, kun mukaan alettiin pakkamaan lehdessä mainittuja MBnetin ilmaisohjelmia – vuosikertojen 1997–1999 painokset veivät kukin peräti neljä levyä. Vaikka MBnetin linjat ja viimeinenkin Apaja-tiedostoarkisto sammuivat vuosia sitten, otos purkista elää näillä kiekoilla.

P.S. MBCD kirvoitti pohdinnan MikroBitin postipalstalle (MB 4/1999): Miksi kukaan enää tilaisi paperilehteä, kun lehdet saa vuosikertalevyllä 150 markkaa paperitilausta halvemmalla ja MBnetin ohjelmatkin tulevat näin kotiin kannettuina, "pientä viivettä" vastaan?

mainituista (ks. artikkelin loppulaatikko).

Itse olisin valmis kantamaan mukanani tietokoneen ja puhelimen lisäksi kolmatta laitetta, joka sähköistäisi kaikki paperitarpeeni tavalla tai toisella. Vielä sellaista ei kuitenkaan ole olemassa. Riittääkö rajallisemman laitteen vetovoima, jos se ratkaisee vain pienen osan laajemmasta paperisiirtymästä? Vastaus lienee hyvin yksilöllinen ja tapauskohtainen. Oma vastaukseni on edelleen: Satunnaisesti. Ei niin usein kuin haluaisin.

Lickliderin loppusanat

Merkittävä osa Business Weekin vuoden 1975 tulevaisuuden toimiston ennustuksista on toki toteutunut. Toimistotyöntekijät ovat siirtyneet paperiröykkiöidensä takaa pääasiassa sähköisten päätteiden äärelle. Sama muutos on tapahtunut tietotekniikalle itselleen, kuten pohdin tämän Skrollin kolumnissani (s. 38). Konekirjoittaminen on siirtynyt sihteereiden ja operaattoreiden työstä lähes kaikkien tehtäväksi.

Vielä reilut 40 vuotta myöhemminkään en kuitenkaan uskaltaisi kutsua toimistoja paperittomiksi. Etenkin muistiinpano- ja vedosteluvälineenä paperi puolustaa edelleen paikkaansa. Enemmistö ihmisistä myös lukee kirjansa edelleen paperilta, eikä kaikkea sisältöä edes saa sähköisenä kaikille keskeisille laitteille. Paperi on jo korvattu sähköisillä ratkaisuilla siellä missä se on ollut helppoa, mutta rakoihin jäänyttä paperisilppua onkin sitten paljon vaikeampi sähköistää. Bell Labsin **Tom Cargill** totesi taannoin koodaamisesta, että ensimmäiset 90 % tehtävästä vie 90 % ajasta ja viimeiset 10 % vaatii sitten toiset 90 % ajasta. Olemmeko viimeisen kympin äärellä myös paperin kanssa?

Joitakin merkkejä edistyksestä on kuitenkin ilmassa, joskin odottamattomissa suunnissa. Korona-aikana yleistyneissä videokonferensseissa muistiinpanojen tekemisen dynamiikka vaikuttaa jo muuttuneen. Useammat naputtavat muistiinpanonsa suoraan tietokoneen näppäimistöllä, koska etäyhteyden päässä katsetta ei ole tarpeen nostaa näytöltä. Käytössä on myös ylimääräinen "kamera" asioiden ikuistamiseen: ruutu- ja videokaappaus. Ehkäpä paperittoman toimiston ennusteissa sorruttiinkin Lickliderin varoittelemaan pitkän tähtäimen muutoksen aliarviointiin. Kenties tulevaisuuden paperittomassa toimistossa ei ole paperia – eikä toimistoa.

Kirjakauppaliiton ja Suomen kustannusyhdistyksen tammikuussa julkaiseman tilaston mukaan viime vuoden lopussa noin 42 % kirjojen euromääräisestä kokonaismyynnistä Suomessa oli sähköistä. Yli puolet tästä luvusta on äänikirjoja. Niiden osuus oli lähes kaksikertainen verrattuna sähköiseen ammattikirjallisuuteen ja peräti viisinkertainen verrattuna tekstimuotoiseen kuluttajakirjallisuuteen. Sähköinen lukeminen lisääntyy eniten siellä missä ei itse asiassa lueta lainkaan.

Metaversumin tai viimeistään teknologisen singulariteetin saapuessa lakkaamme sitten myös kirjoittamasta.

Kuva: Janne Sirén

Linkit artikkelin mainintoihin ja maksuttomat pdf-versiot yli vuoden vanhoista Skrolleista verkkojatkoilla: skrolli.fi/numerot.

Lukeminen kannattaa aina

Sähköiset kirjat sähkömustenäytöllä ovat olleet itselleni arkipäivää jo toistakymmentä vuotta. Ne ovat lisänneet lukemistani huomattavasti, koska uusia kirjoja on helppo hankkia. Pääasiallinen lukulaitteeni on Amazon Kindle (amazon.com/kindle) eri muodoissa, mutta olen käyttänyt myös muita e-lukijoita, kuten Kobo Aura H2O:ta (kobo.com), jossa on enemmän ominaisuuksia. Tunnistan toki, että kielialueesta ja alustasta riippuen sähkökirjojen saatavuudessa on paljon eroja – kun tarve kohdistuu suomenkieliseen kirjaan, saatan joutua tyytymään paperiin tai turvautumaan konversiotyökaluihin (ks. calibre-ebook.com). Mutta koska luen paljon englanniksi, yleensä etsimäni kirja on saatavilla sähköisenä. Suomi-tarjontakin kasvaa kaiken aikaa.

Paperille painetun yksivärisen romaanitekstin korvaaminen sähkömusteella onkin verrattain helppoa. Siirtyminen fyysisestä digitaaliseen ei kuitenkaan ole aina näin suoraviivaista. Esimerkiksi aikakauslehden siirtäminen paperilta digiulottuvuuteen on haastavampaa, ellei siirrytä puhtaaseen verkkosivujulkaisuun. Aikakauslehdillä ja vastaavan tyylisillä tietokirjoilla on julkaisumuotoina edelleen etunsa, mutta niiden paperille suunnitellut näköisjulkaisut istuvat huonosti tietokoneiden tai tablettien ruuduille. Silmät myös väsyvät lcd- ja led-värinäyttöjen paistatuksessa, ja värillinen sähkömuste on vielä lapsenkengissään. Muna vai kana -ongelmana monet tahot ovat myös mananneet sähköisten aikakauslehtien pientä kysyntää; Skrollin edesmenneen iOS- ja Android-digilehden myyntilukuja voi kauhistella Postipalstalla (s. 84).

Toisaalta koronapandemian kiihdyttämät logistiikkaongelmat ja komponenttipula, joka on näkynyt puutteina jopa paperin saatavuudessa, ovat lisänneet sähköisen jakelun houkuttelevuutta. Esimerkiksi edellisessä Sirénin sisäpiirissä mainitut MEGA65-retrotietokoneen ensitoimitukset viivästyivät joulukuulta maaliskuulle osin pahvilaatikoiden saatavuuden takia. EU-alueen tullausvaatimusten kiristyminen viime kesänä sekä Suomen kolmatta vuotta jatkuva postikaaos ovat lisäksi pahentaneet paperijakelun tilannetta pandemiasta riippumattakin.

Kun yksinkertainen kirjekuori läntisestä Euroopasta voi olla matkalla Suomen osoitteeseen puolitoista kuukautta ja kotimaistenkin lehtien jakelupäivä venyy viikoksi tai kahdeksi, sähköpostiin kilahtava pdf-lehti tai muu latauslinkki alkaa kuulostamaan hyvältä.

Sijainnin tasa-arvoa

Paperista ja yleisemmin fyysisistä jakelumedioista luopuminen on kuitenkin paitsi käytännön kysymys, myös vahvasti tunneasia. Koska luet tätä artikkelia todennäköisimmin Skrolli-nimisestä paperilehdestä – paperi- ja digimyyntimme suhde on noin 100:1 – sinulla on epäilemättä itselläsikin jonkinlainen näkemys aiheesta. Tietotekniikan retromarkkinat ovatkin eräs prisma, jonka lävitse teemaa voi tarkastella. Fyy-

Kuva: Janne Sirén

Keskellä Skrollin Android-digilehti Boox Nova3 Colorin värillisellä 7,3 tuuman sähkömustenäytöllä. Vasemmalla väriverrokkina Samsung Galaxy Z Fold3 7,6 tuuman taittuvine AMOLED-näyttöineen sekä kokovertailuna pari A4-kokoista Skrollia. Reilun 7 tuuman näyttö ei ole vielä aivan riittävä tämän kokoisen aikakauslehden lukemiseen ilman suurennusta, mikä on hitaan sähkömusteen yhteydessä epäkäytännöllistä.

siset painotuotteet ja hypisteltävät ohjelmapaketit kun ovat tässä yleisössä edelleen toivottuja ja voimissaan.

Moni retrotuotteiden hankkija haluaa fyysisen keräilykappaleen. Koska retrotarjonta tulee pääasiassa pieniltä toimijoilta, posti on usein ainoa realistinen vaihtoehto tällaisten tuotteiden toimittamiseen – riittävää volyymia kansainvälisten kuriirifirmojen kustannustehokkaaseen käyttämiseen ei synny. Moni retrotilaaja odottaa kuitenkin yhtä lailla myös pakkauksen sisältöä ja haluaisi sen mieluummin aikaisemmin kuin myöhemmin. Sähköinen versio onkin tässä ennen kaikkea tasa-arvoistava tekijä, sillä kaikilla tilaajilla on jotakuinkin yhtäläinen mahdollisuus ladata se samaan aikaan, sijainnistaan ja postin kulusta riippumatta. Tarjoamalla digitaalisen kopion lähettäjä tekee voitavansa toimitusketjun tehostamisessa kaikille.

Kokemukseni mukaan suuri osa retrotuotteita julkaisevista tahoista tarjoaakin jo nyt sähköisen kopion, yleensä pdf-version paperituotteen lisäksi tai latauslinkin fyysisen ohjelmatilauksen yhteydessä. Sähköinen versio vapautetaan jakeluun joko heti ilmestymispäivänä tai vaihtoehtoisesti, kun ensimmäiset fyysiset kappaleet ovat tulleet perille. Esimerkiksi edellisissä Sirénin sisäpiireissä (esim. Skrolli 2021.3, 2021.4) esitellyt Fusion Retro Booksin tietokonelehdet saa Skrollin tapaan paperitilauksen yhteydessä ladattua myös pdf-tiedostoina (ja kansilevykkeet näköistiedostoina).

Luonnollisesti myös puhtaasti sähköinen tilausvaihtoehto on tällöin luonteva ja ekologinen lisuke tarjontaan. Sen suosion voi ennakoida vain kasvavan ajan mittaan. Yllättävän moni taho tosin jakelee sähköisiä versioita jopa täysin maksutta – tämä on tavallista esimerkiksi Commodore 64 -retropelien markkinoilla, ja Skrollikin vapauttaa noin vuoden ikäiset pdf-lehdet maksuttomaan jakeluun (skrolli.fi/numerot).

Digitaalista epäröintiä

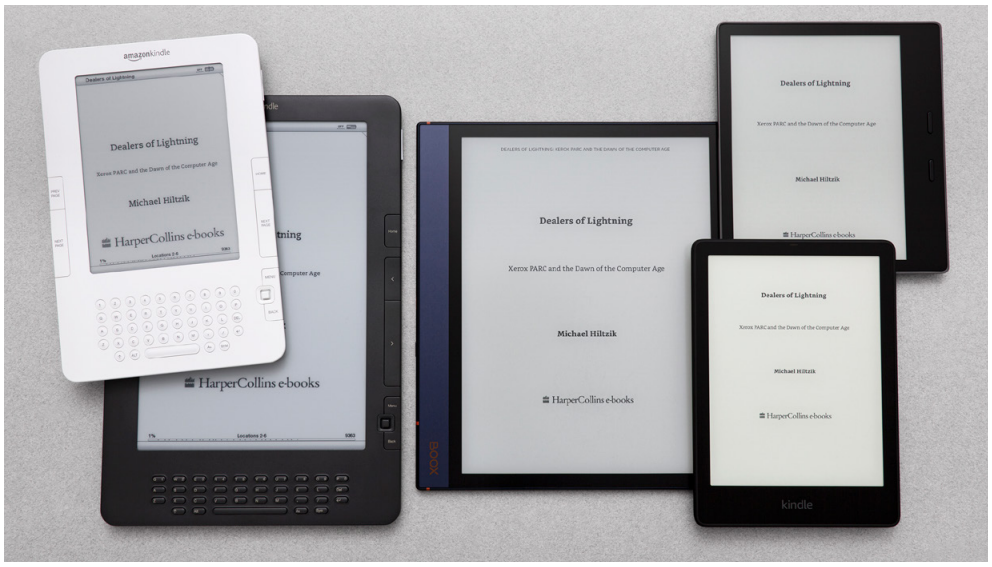
Olen kuitenkin törmännyt myös lukuisiin digitaalista jakelua vastustaviin reaktioihin. Skrollin piirissä esimerkiksi postiviiveiden takia aikaistamamme pdf-lehden jakelu on herättänyt kriittistäkin palautetta. Jotkut lukijamme toivoisivat edelleen, että pdf-lehti saapuu vasta, kun paperilehti on tullut jotakuinkin kaikille postissa. Vaikka huomioisimme vain keskimääräiset Suomen sisäiset postitusviiveet, tämä tarkoittaisi pdf-lehden ilmestymistä vasta noin viikko paperijakelun aloittamisen jälkeen.

Esimerkkejä on muitakin. Pari vuotta sitten yritin vakuutella ZX Spectrum Next -retrotietokoneen tekijätiimiin kuuluvia pelijulkaisijoita fyysisten ja digitaalisten hybridijulkaisujen meriiteistä – tuloksetta. Heidän ajatuksenaan oli ilmeisesti suojella nautintoa fyysisen kappaleen saapumisesta, ja kenties myös varjella fyysisten pelien myyntiä. Pian tämän jälkeen saimme

ihmetellä, kuinka Kreikasta lähetetyillä postisäkillisillä retropelejä kesti kuukausia löytää perille keskellä nousevaa koronamyrskyä. Yhteisen pelihetken sijaan sosiaalinen media täyttyi vaihtelevan mittaisesta odotuksen tuskasta.

Toinen esimerkki on mainio englantilainen retrolehti FREEZE64, jonka tekijää **Vincenzo "Vinny" Mainolfia** haastattelin kaikkein ensimmäisessä Sirénin sisäpiirissä Skrollissa 2018.3. FREEZE64 ilmestyy pääasiassa vain paperilla, joskin tekijä on myynyt yksittäisiä pdf-kappaleita muutamista numeroista (räätälöitynä ostajan nimellä). Kun ehdotin hiljattain jatkuvaa pdf-versiota paperilehden rinnalle jakelua nopeuttamaan, Vinny vastasi suorasukaisesti: "FREEZE64:ää ei ole luotu pdf-dokumentiksi, enkä ole aikeissa muuttaa sitä sellaiseksi. Mieluummin lopettaisin kuin alkaisin myymään jokaista numeroa pdf:nä." Sääli. Huomaan itse lukevani lehtien pdf-versioita yllättävän usein, kun fyysinen kappale on vielä matkalla tai se pitäisi vaivautua kaivamaan esiin.

Eläköitynyt Amiga-johtaja **Petro Tyschtschenko** oli samoilla linjoilla saksankielisen muistelmakirjansa kanssa, josta kerroin kolumnissani Skrollissa 2019.4. Petro mainitsi minulle eräällä automatkalla Saksassa toivovansa, että ihmiset saavat laadukkaan kirjan, jonka he voivat laittaa esille kirjahyllyyn. Hän tuntui kantavan huolta sähköisen kirjan katoavaisuudesta. Vaikka huoli on ymmärrettävä henkisen perinnön kannalta ja Petron paperikirja onkin oikein hieno, en ole vakuuttunut, että päätös oli oikea. Koska englanninkielinen käännös on jäänyt ainakin toistaiseksi kustannus- ja kiinnostussyistä tekemättä, vain paperilla oleva saksankielinen kirja rajoittaa sanojen tavoittavuutta. Sähkökirja olisi helpompi konekääntää, ja tietyssä mielessä bitit ovat myös ikuisia.



Kuva: Janne Sirén

Amazon oli sähkömustetuotteiden edelläkävijä ja popularisoija, jolla oli erikoisempiakin malleja, kuten 9,7-tuumainen Kindle DX Graphite (2010) vasemmalla alhaalla. Vasemmalla ylhäällä on ensimmäinen euro-Kindle: Kindle 2. Nykyisin yhtiö on keskittynyt peruslukulaitteisiin ja innovaatiota on haettava muualta. Oikealla on uusimmat Kindlet, sivunvaihtonäppäimellinen Oasis 3 (2019, ylinnä) ja Paperwhite 5 Signature Edition (2021), jonka suurin uudistus on micro-USB:n vihdoinkin korvannut USB-C-latausliitin sekä langaton Qi-lataus. Yksi innovaattori on Onyx, jonka Boox Note Air2 on keskellä. Muistiinpanotoimintojen lisäksi Kindle-kirjat toimivat siinä Androidin Kindle-sovelluksella. Kindle-laitteet ovat kuitenkin edelleen hyvä valinta – ja oma valintani – romaanien lukemiseen. Kindlejä myy Suomeen muun muassa Amazon.co.uk.

Monenlaisia tapoja

Jokainen julkaisija saa tietenkin tehdä omat valintansa, kuten myös asiakkaat. Itse suosin retromarkkinoilla usein tuplavaihtoehtoa: Fyysinen keräilykappale hyllyyn, ihailtavaksi ja hitaisiin nautiskeluhetkiin sekä verkkoladattava kopio välittömään käyttöön, konekääntämiseen, varmuuskopioksi ja emulaattoreihin ladattavaksi. Olen valmis maksamaan tästä hieman ylimääräistäkin, joskin pidän molempien tarjoamista samaan rahaan järkevänä. Tarjoaisin itse asiakkaille aina myös mahdollisuuden ostaa tai saada tuotteista pelkkä sähköinen versio, sillä materian kerääminen on yhä vähemmän muodissa – minullakin toisissa tuoteryhmissä. Tämän filosofian mukaan toimimme myös Skrollissa.

Sähköistä julkaisua voi toki lähestyä monella tavalla. Esimerkiksi pdf-tiedostojen muutoksiin suhtaudutaan eri tavoin. Fusion Retro Booksin lehdet, kuten Zzap! ja Crash, saavat pdf-julkaisunsa jo hieman ennen lehden menoa painoon. Nopeimmat lukijat ehtivät tällöin ilmoittaa virheistä, jotka korjataan paperiversioon. Skrollissa homma on toiminut toisin päin: pdf-versioon saatetaan tehdä pieniä oikaisutason korjauksia vielä painoon menon jälkeenkin, mutta ennen lehden ilmestymistä, eli noin kahden viikon ajan lehden valmistumisesta. Skrollin juuri alkaneen kymmenennen vuosikerran kunniaksi kerrattakoon, että alkuaikoina yksi numeromme oikaistiin myös "kräkkäämällä" se ilmestymisen jälkeen. Lataa skrolli.fi/skrolli.2013.2-proper_cracked_by_byterapers.zip ja löydä korjaus.

Olemme Skrollissa jaelleet pdf-lehteä paperitilaajille ilman lisähintaa perustamisestamme asti sekä tarjonneet erillisiä pdf-tilauksia, mutta tämäkään sähköinen julkaisumuoto ei ole kaikkien mieleen. Niinpä meillä oli myös vaihtoehto: Skrolli-mobiilisovelluksessa julkaistu digilehti (2017–2021, kuvassa s. 33). Sovelluksella olikin muutamia uskollisia asiakkaita, jotka kokivat pdf-lehtien lukemisen vaivalloiseksi. Heitä oli kuitenkin aivan liian vähän tappiollisen ja työlään palvelun jatkamiseen. Vuosi sen jälkeen, kun uusien sovellustilausten myymisestä luovuttiin, eräs pdf-lehtitilaukseen vaihtanut asiakas ilmoitti, ettei ollut saanut aikaiseksi lukea ainuttakaan korvaavista pdf-lehdistä. Hän ilmoitti palaavansa tilausta uusiessaan paperilehteen.

Toisinaan pdf-tiedostoakin voi kuitenkin tulla ikävä. Itselleni näin on käynyt sähköisten koulukirjojen kohdalla. Sähköisten kirjojen käyttö lisääntyi hiljalleen Suomen kouluissa viime vuosikymmenen aikana ja viimeistään vuoden 2021 oppivelvollisuusuudistus – joka takaa maksuttomat oppikir-

jat myös toisen asteen opiskelijoille – vauhditti digiloikan lentoon. Suomen kustannusyhdistys kertoi tammikuussa, että 29,3 prosenttia viime vuoden oppimateriaaleista toimitettiin sähköisenä. Eikä siinä mitään, mutta ne kymmenien klikkausten ja frame-ristikon taakse sisältönsä piilottavat verkkosivukirjat ovat kamalaa päntättävää. Valtakunta näköisjulkaisusta.

Digikirjoittamisessakin on nurjat puolensa. Kun tehtävätkin tehdään tietokoneella, käsin kirjoittamisen muistijälki jää syntymättä ja etenkin matematiikan kohdalla aikaa tuhrautuu käyttöliittymäleikkeihin.

Pdf-tiedostoissa on myös eräs aivan olennainen etu: riippumattomuus arvaamattomista sähköisistä julkaisualustoista.

Testissä e-ink-Androidit

Yksi hidaste digitaalisten sisältöjen omaksumiselle ovatkin pirstaleiset ja suljetut jakelualustat. Vaikka mieluisa laite sisältöjen käyttöön löytyisikin, ei ole lainkaan taattua, että haluttua sisältöä olisi sille saatavissa – ainakaan missään käyttökelpoisessa muodossa. Yksi tässä mielessä lupaava jakelutie ovat Android-pohjaiset e-ink-tabletit, sillä niissä yhdistyvät sähkömusteen edut sekä mahdollisuus käyttää epub- ja pdf-tiedostojen lisäksi esimerkiksi Google Play -kaupan Android-sovelluksia ja modernia internet-selainta.

Android-tablettienkin kanssa ollaan toki pohjimmiltaan sisältöalustojen armoilla, mutta tällaiset tuotteet kytkeytyvät sentään poikkeuksellisen laajaan kirjoon eri tarjoajia. Yksistään Google Play -kauppa avaa ovet moniin sellaisiin lukukokemuksiin, joihin perus-Kindle ei taivu, ja laitteet tukevat yleensä myös sovellusten ja sisältöjen sivusta lataamista. Yksi tällaisia Android-tabletteja valmistaja yritys on kiinalainen Onyx, joka on kymmenen vuoden aikana tuottanut hengästyttävän kirjon erilaisia vaihtoehtoja Boox-tuotemerkkinsä alle (euroshop.boox.com).

Testasin kahta Onyxin uutuutta tätä palstaa varten: Boox Note Air2 (hinta alkaen 500 €) on A5-kokoisella 10,3 tuuman e-ink-näytöllä varustettu mustavalkotabletti ja Boox Nova3 Color (400 €) 7,8-tuumainen, kolmannen sukupolven E Ink Kaleido Plus -värisähkömustenäyttöinen tabletti. Molemmat ladataan USB-C:stä ja niissä on sekä WLAN- että Bluetooth-yhteydet. Laitteiden pääasiallinen ero on näytössä ja koossa. Onyxin tarjontaan kuuluu lisäksi 13,3-tuumaisia eli A4-kokoisia e-ink-mustavalkotabletteja ja Mira-merkkisiä sähkömustenäyttöjä tietokoneille.

Testaamissani Boox-tableteissa on etuvalo ja kosketusnäyttö sekä mahdollisuus paineherkän osoitinkynän käyttöön, joten niillä on yhtymäkohtia myös reMarkablen tuoteryhmään. Laitteissa on myös mahdollisuus käyttää Google Play -sovelluskauppaa, joskin kaupan käyttöönotto vaatii pientä säätöä.[1] Google Playn myötä Boox-tableteilla on mahdollista lukea vaikkapa Kindle-kirjoja Kindle-sovelluksessa. Myös viime vuonna tarjolla ollut Skrolli-sovellus digilehtineen toimi moitteetta.

Kun sanon moitteetta, tarkoitan teknisesti moitteetta. Boox-tabletit toimivat kuin mikä tahansa Android-älylaite. Ne ovat myös sangen siroja ja tyylikkäitä, enkä moittisi niiden suorituskykyäkään. Snapdragon 636 (8 ydintä, 1,8 GHz) -pohjaiset Booxit ovat esimerkiksi huomattavasti tehokkaampia kuin reMarkable 2 (A9, 2 ydintä, 1,2 GHz). Kehuja hillitsee kuitenkin sähkömustenäytön luonne. Sähkömuste liimattuna perinteisemmille näytöille tarkoitetun Androidin päälle ei nimittäin solju lainkaan yhtä sutjakkaasti kuin reMarkablen tai Kindlen tarkoitukseen räätälöidyt käyttöliittymät. Sanalla sanoen Boox-kokemus on vähän kökkö.

Onyx ei myöskään ole keskittynyt muistikirjan korvaamiseen reMarkablen tapaan, joten jälkimmäinen on selkeästi parempi käsimuistiinpanojen tekemiseen, ellei erityisesti tarvitse Booxin suurinta näyttökokoa tai etuvaloa. Toisaalta Boox ei vaadi maksullista pilvitilausta täysipainoiseen toimintaan. Muutakin edistystä tapahtuu: Uusimmat Booxit, kuten Note Air2, tarjoavat karkeamman näyttöpinnoitteen reMarkablen tapaan. Tämä tekee kirjoitustuntumasta hieman paperimaisempaa. Käsin kirjoitettaessa on yllättävän tärkeää, ettei kirjoituspinta ole liian liukas.

Booxit ovatkin sinänsä aivan toimivia laitteita, jos nimenomaan tarvitsee sähkömusteen ja Google Play -sisällön yhdistelmää. Jos tarvitsee vain sähköistä muistiinpanoalustaa, reMarkable 2 on parempi. Se tuntuu myös paljon Note Air2:ta kevyemmältä, vaikka painoeroa pitäisi olla vain 17 grammaa. Vaaka paljastaakin fuskauksen: reMarkable oli vaa'allani 7 grammaa speksejään kevyempi (396 g, ei 403 g), mutta Note Air2 peräti 20 grammaa speksejään painavampi (440 g, ei 420 g).

Itselleni kaikkein kiintoisin kysymys oli kuitenkin se, olisiko värillisestä Boox Nova3 Colorista aikakauslehtien lukemiseen.

Värillinen sähkömuste

Kaleido Plus on E Ink -yhtiön uusin värinäyttötekniikka, joka korvasi vuoden 2019 Kaleidon viime vuoden alussa. Nova3 Color edusti siis testihetkellä alan terävintä kärkeä. Ammattituotteista löytyy vieläkin parempaa sähkömustetta, mutta niiden päivitysnopeus ei riitä e-lukijoihin. Kuten Kaleido, Kaleido Plus perustuu mustavalkoisen sähkömustenäytön päälle asennettuun muoviseen värisuodattimeen. Nova3:ssa e-ink-näytön tarkkuus on 1872×1404 (300 dpi) ja värisuodattimen tarkkuus 624×468 (100 dpi). Käytännössä matalampi väritarkkuus ei kuitenkaan ollut itselleni ongelma – 100 dpi on aivan riittävästi. Suurempi ongelma on se, että värejä on vain 4096 ja harmaansävyjä 16, mutta nämä rajoitukset ovat sähkömusteelle tavanomaisia piirteitä.

Hyvänä ja huonona puolena onkin todettava, että värillinenkin sähkömuste on ennen kaikkea sähkömustetta. Kaleido Plus -näyttö toimii hyvin kirkkaassa auringonvalossa ja rasittaa silmiä perinteistä näyttöä vähemmän. Sitä voi käyttää myös ilman etuvaloa. Toisaalta näytön kuva on vaisu ja päivittyminen aiheuttaa kovasti haamukuvia, joiden kanssa tasapainoisten asetusten löytäminen ottaa aikansa ja riippuu sisällöstä. Onneksi Booxeissa saa pakotettua ruudunpäivityksen

[1] *Google Play -vipu tulee ensin asettaa Enable-tilaan Boox-tabletin Application- tai Apps / App Management -asetuksista (firmwaresta riippuen). Play valittaa, ettei laite ole rekisteröity tai Google Play Protect -sertifioitu. Tämä on normaalia. Rekisteröinti tapahtuu hakemalla "GSF ID" -tunniste samasta asetusvalikosta. Kestää tunneista päivään, että Play alkaa toimia. Jos rekisteröinti ei ota aktivoituakseen, käännä Google Play -vipu pois päältä ja takaisin päälle.*

pyyhkäisemällä alareunasta ylöspäin – tälle tulee Androidin kanssa tarvetta.

Entäpä ne värit? Vähäisen määrän lisäksi sähkömustenäytön värit ovat haaleita ja enemmänkin viitteenomaisia kuin valokuvamaisen tarkkoja. Ne toimivatkin paremmin sarjakuvissa tai kaavioissa kuin valokuvissa. Värejä ja näytön etuvalon kirkkautta voi säätää useammalla tavalla, mutta tarjonta muistuttaa kaikilla asetuksilla mietoa pastellikeitosta. Värisisältöjä katsellessa selkeysero mustavalkonäyttöön verrattuna on silti merkittävä, koska tekstilaatikoiden taustavärit eivät sotkeennu tekstiin. Tämä on tärkeä edistysaskel värillisten taittojen luettavuuden kannalta.

Kulahtaneet ja vähäiset värit eivät lopulta ole kummoinenkaan ongelma. Ne ajavat asiansa aivan hyvin vaikkapa Skrollin lukemisessa. Suurin ongelma on näytön värisuodattimen rakenne, joka peittää koko kuvan himmeään raidoitukseen. Raidoitus muistuttaa vanhojen CRT-näyttöjen juovaisuutta ja peittää allensa valitettavasti myös suodattimen alla olevan mustavalkonäytön, riippumatta siitä onko värisisältöä esillä. Niinpä mustakin sähkömuste menettää terävimmän teränsä ja sen mukana osan paperimaisesta taikuudestaan. Syy on Kaleido Plus -näytössä, ei Booxin tabletissa. Mustavalkoinen Note Air2 ei esimerkiksi kärsi tästä.

E-ink-Skrolli

Nova3 Colorin ja Note Air2:n Snapdragon 636:ssa on enemmän kuin riittävästi tehoa pyörittelemään esimerkiksi Skrollin pdf-lehtiä sisäänrakennetussa pdf-lukijassaan. Itse asiassa myös reMarkable 2 ja uusi Kindle Paperwhite 5:kin selasivat Skrollin pdf:ää aivan mainiosti.

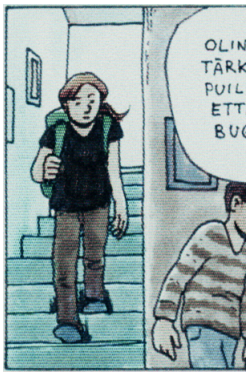
Mutta koska sähkömustetta ei ole käytännöllistä zoomailla millään laitteella, sivu pitää mahduttaa kerralla ruudulle. Nova3 Colorin 7,3 tuuman näyttö on lopulta aavistuksen liian pieni täysikokoiselle aikakauslehdelle. Jos näyttö olisi neljänneksen suurempi tai värinäytön teksti saman verran tarkempaa, tämä voisi riittää. Tällaisenaan laite lienee parhaimmillaan sarjakuville, joissa sivut ovat kenties pienempiä ja teksti ainakin suurta.

Värittömyyskään ei välttämättä ole este. Jopa uuden Kindlen pdf-toisto on jo laadullisesti aivan mainiolla tasolla, mutta A4-kokoinen sisältö jää sen 6,8 tuuman näytöllä on liian pieneksi. 10,3 tuuman mustavalkotabletit sen sijaan pärjäsivät aikakauslehtien kanssa paremmin. Note Air2:n ja reMarkable 2:n samankokoiset ja -tarkkuiset näytöt ovat olennaisesti Nova3 Coloria selkeämpiä – riittävä harmaasävylaatu yhdistettynä tarkkaan ja kookkaaseen näyttöön toimii. ReMarkable 2:n perus-pdf-toisto yltää hienoisesti laatuykköseksi, joskin Note Air2 omaa enemmän kustomointivaraa.

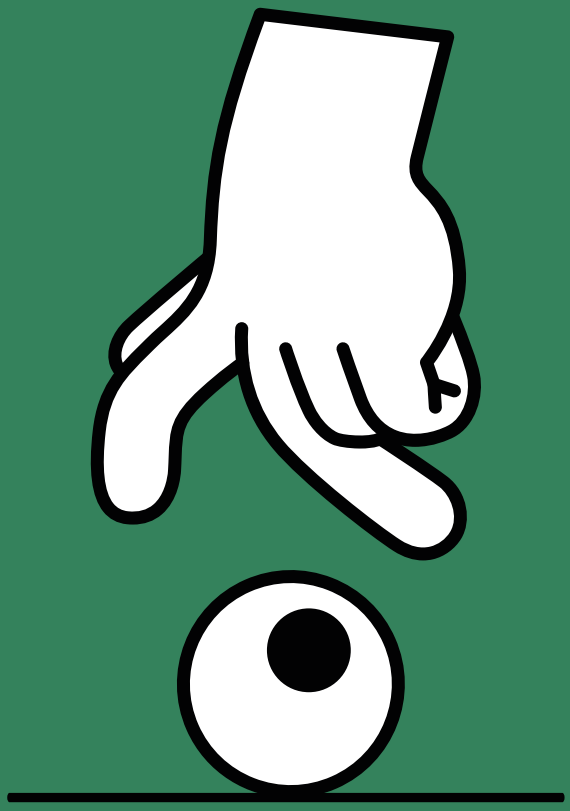
Vaikka reMarkable 2:n ja Booxien perusnäytöissä on paljon samaa, eroa on myös siinä, mitä näytön päälle on kasattu. Koska reMarkablessa ei ole etuvaloa tai värisuodatinta, sen sähkömuste on häiriöttömästi lähes näytön pinnassa, kuten siinä ensimmäisessä valottomassa e-lukijassani reilu vuosikymmen sitten. Vain kosketuskalvo on tiellä, mikä mahdollistaa harvinaisen puhtaan sähkömustekokemuksen. Taika on tallella – ainakin kunnes joku sammuttaa huoneesta valot.

E Ink on saamassa kilpailijan värillisen sähkömusteen saralla: Display Electronic Slurry -sähkömustenäyttö perustuu hiukkaskapselien sijaan patomaiseen rakenteeseen, jonka sisällä sähkömuste elää. DESissäkin värit on kuitenkin toteutettu erillisellä suodattimella. Viime vuonna julkistettu Reinkstone R1 (reinkstone.com) on ensimmäinen näyttöön perustuva Android-tabletti, joskin kirjoitushetkellä sen toimitukset ovat viivästyneet komponenttipulan vuoksi. Ennakkotestien mukaan eroa e-inkiin on jonkin verran, mutta niin hyvässä kuin pahassa. Vallankumousta saa vielä odottaa.

Värillinen sähkömuste on toistaiseksi vahvimmillaan sarjakuvissa – tässä Skrolli 2021.4:n sivu 47 sähkömusteena. Kuvissa 10,3" reMarkable 2 vasemmalla (alhaalla) ja 7,3" Boox Nova3 Color keskellä (ylhäällä). Kuvapari on ladattavissa verkkojatkoilta: skrolli.fi/numerot.

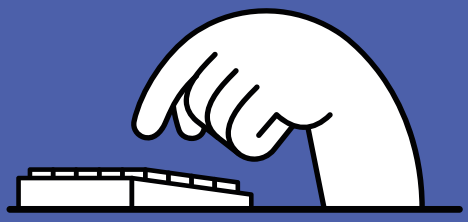


Kuvat: Janne Sirén / Sarjakuva: JJ Nääs, Skrolli 2021.4



Fork(); Pull >Merge (Push)

The podcast for developers — made by developers.



...	
s2e4	Let's Map This Out! Working with GIS and OpenStreetmap Data
s2e3	Curiosity Drive or Why I Moved from the US to Finland
s2e2	Do You Really Need Artificial Intelligence in Your Software?
s2e1	Code and Emotion. Why Anthropology is Important for Creating Good Digital Products
s1e8	Deep Dive into Advanced TypeScript
s1e7	Clojure and Its Superpower, The REPL
s1e6	Category Theory for the Non-PhD – and What to Use it for
s1e5	Finding the Corner of the Internet You Love with Amsterdam Hackers
s1e4	The Promise of Tasks: Concurrency in a Single-Threaded World
s1e3	Time to Ditch the Boilerplate: Safe, Fast & Easy Data Manipulation with Statically Typed Optics
s1e2	GraphQL is the God Mode of Web Development
s1e1	Will Harmaja be the React Killer We've Been Waiting for?

Season two is out now — listen to new episodes bi-weekly

Listen on Apple Podcasts Spotify Listen on Google Podcasts

reaktor.com/forkpullmergepush

Reaktor

Printtiä

Vanhan sanonnan mukaan maantietä viilettävän, tallennusmedioita täynnä olevan rekan tarjoamaa kaistanleveyttä ei kannata aliarvioida – rekan tiedonsiirtonopeus saattaa yllättää. Samanlainen kapasiteettiyllättäjä voi löytyä lähimmästä paperikorista.

Janne Sirén

Kuulin rekkavertauksen ensimmäisen kerran opiskellessani Helsingin yliopistolla 90-luvulla. Hätäisempi voisi kuvitella, että tilanne on sittemmin muuttunut, mutta sitä mukaa kun tietotekniikan käyttö ja kapasiteetit ovat kasvaneet, myös tallennetun tiedon määrä on lisääntynyt. Niinpä esimerkiksi Amazonilla on edelleen Snow-niminen tuoteperhe fyysisiin tiedonsiirtoihin. Suurin tarjolla oleva palvelu on Snowmobile, sadan petatavun datakuormia siirtelevä konttirekka. Oma 1 Gbps:n kuituyhteyteni siirtäisi tätä tietomäärää 25 vuotta – siinä kannattaa jo tilata datataksi.

Data liikkui yllättävän tehokkaasti myös toisella tavalla: paperilla.

Gigabitti paperilla

Törmäsin Amazonia pienimuotoisempaan, mutta silti merkitykselliseen esimerkkiin tallennuskapasiteetista 90-luvun puolivälissä. Tämä tapahtui opiskeluaikaisessa työpaikassani MikroBitti-lehdessä. Lehden toimituksessa keskusteltiin vanhojen MikroBitti-taittojen arkistosta. Lehteä oli tässä vaiheessa taitettu digitaalisesti jo joitakin vuosia, mutta taittotiedostoilla ei ollut minkäänlaista yhteiskäyttöistä arkistoa ainakaan ennen vuotta 1996 – jos oli arkistoa lainkaan.

Tai olihan sellainen, kuten toimitussihteeri **Riitta Tumppila** muistutti. Vanhat taitot oli lehden perustamisesta lähtien arkistoitu kovakantisiin kirjoihin, joihin oli niputettu kunkin vuoden painetut paperilehdet. Kyse ei ollut pelkistä kansioista, vaan ammattimaisesti sidotuista kirjoista (esim. Sitomotyö Oy, Järvenpää). Tämä oli alan tapa. Kolumnikuvassa ovat C=lehden vuosikirjat – vastaavat kirjat koristivat hyllyjä ympäri MikroBitin toimitusta, ja digitalisoidussa muodossa nämä arkistolehdet muodostavat tänä päivänäkin julkaisun Retrobitti-osaston verkossa.

Kyse ei tietenkään ollut pelkästä tilasta – lehtiä taitettiin pitkään mekaanisesti ja pdf:n kaltaisen universaalin taltiointimuodon yleistyminen otti sekin aikansa – mutta laskelma osoittaa, että tilallakin oli väliä. Pakkaustasosta ja laskentatavasta riippuen 80-luvun MikroBitti-vuosikirjassa on reilun gigatavun verran tietoa. Kymmenen vuosikertaa vie siis yli kymmenen gigatavua. Määrä ei olisi mahtunut tuon aikaisten työasemien kiintolevyille, saati muistiin tai levykkeille, ja CD-ROM-aikakaudellakin oli vielä tyytyminen karsittuun tarkkuuteen.

Toimituksen kirjahyllyille sen sijaan mahtui useita "painolaatuisia" kopioita.

Vektoreita ja johtimia

Tietokonenuoruuteni 80-luvulla ja 90-luvun alussa oli sangen yleissivistävä. Omien ja ystävien runsaslukuiset kotikoneet sekä vanhempien työkoneet avasivat 8- ja 16-bittisten kotimikrojen sekä PC:n maailmat. Lisäksi olin vuosia mukana parissa kerhossa, joissa tutustuin varhaisiin Macintosh-sukupolviin, Nokian teollisuus-ATK:hon, radio- ja verkkoteknologioihin sekä Unix-järjestelmiin tyhmistä päätteistä Sun Sparcstationeihin. Urani alkupään työtehtävät mainframe-tuotteita edustaneen yhtiön hommissa avasivat nekin silmiä tietotekniikan kirjolle.

Selviä aukkoja jäi silti. Vaikka tuolloin käyttämäni joukko tietokoneita 70-luvun lopulta 90-luvulle olikin monipuolinen sekä raudaltaan että käyttöjärjestelmiltään, joissain asioissa kyseessä oli homogeeninen ryhmä. Esimerkiksi kaikissa koneissa oli tekstiin ja pikseleihin kykenevä näyttö (tai näytöllinen etäyhteys) sekä magneettiset tallennusmediat – nämä olivat vakiintuneita tietokoneen ominaisuuksia. Aina näin ei kuitenkaan ole ollut. Edeltävinä vuosikymmeninä tietokoneissa ei välttämättä ollut näyttöä lainkaan, eikä aina magneettisia medioitakaan.

Jos näyttö olikin, sen toimintaperiaate saattoi olla nykymittapuulla vieras. Monet varhaiset tietokonenäytöt olivat nimittäin niin kutsuttuja vektorinäyttöjä. Oskilloskoopeistakin löytyvässä vektorinäytössä kuva piirretään koordinaattien välisinä viivoina, mikä vaatii pistekuvaa vähemmän resursseja. Vektorinäyttö ei kuitenkaan ole kovin optimaalinen tekstikäytössä, sillä teksti on esitettävä hieman kömpelöinä ja pahimmillaan vilkkuvina viivapiirroksina. Toisaalta viivagrafiikassa näyttötyyppi on omaa luokkaansa. Viivojen laatu haastaa monia nykyaikaisiakin pikselinäyttöjä.

Varhaisissa tietokoneissa sovellettiin myös monenkirjavia fyysisiä tallennusmedioita. Kun vektorinäytön kuva haluttiin talteen, järjestelmään saatettiin liittää erillinen laite kuvan siirtämiseksi filmille – varsinainen *screenshot* siis – tai plotterille, joka kirjaimellisesti veteli vektorit paperille tussilla. Data saattoi olla myös elektroniikkaa: Suomen ensimmäisen ydinvoimalan simulaatiomalliin liittyviä laskelmia tehtiin osin analogisella tietokoneella, jonka "ohjelmat" olivat

piuhoina ja komponentteina vaihdettavilla kytkentälevyillä. Tällaisen laskennan tietovirrat olivat epäilemättä paikoin niin suuria, ettei niiden käsittely täysin digitaalisena olisi ollut tuohon aikaan käytännöllistä.

Myös varhaisia IBM:n reikäkorttikoneita ohjelmoitiin kytkentälevyillä eli vetämällä fyysisiä johtimia liittimestä toiseen (yksi kytkentäkuva näkyykin sivulla 3). Näiden koneiden tulostusmerkistö oli puolestaan vaihdettavilla, metallisilla kohokuvakorteilla, jotka kytkivät tulostusprosessin aikana tulostimen johtimia. Metallilaatat sisälsivät eräänlaista kaiverrettua pistekirjoitusta. Kieltä tai fonttia vaihdettiin vaihtamalla tämä postimerkin kokoinen metallin pala. Laatat olivat käytännössä fyysistä ROM-muistia. Veteraanit kertovat, että ääkkösetkin saatiin tulostettua, kunhan oikea kortti oli koneessa.

Näyttömuisti

Suurin ATK-historian aukkopaikkani liittyi kuitenkin paperin rooliin tietotekniikassa. Mietin nuorempana vanhoja valokuvia katsellessani: miten tietokonetta käytettiin ilman näyttöpäätettä? Arkistokuvissa huomio kiinnittyy tietysti koneiden näyttäviin valo- ja kytkinpaneeleihin, joita kyllä hyödynnettiin vianetsinnässä ja ohjelmien syöttämisessä konekielikäsky kerrallaan. Mutta vaikka tietokoneajot olivat automaattisen tietojenkäsittelyn etsikkoaikana eräajoluontoisempia ja sisälsivät vähemmän interaktiota, jotain tuntui silti puuttuvan. Pelkillä vivuilla ja valoilla ei voi pötkiä pitkälle.

Arkistokuvista ei näkynyt, että tietokoneita käytettiin pitkälti paperilla. Ohjelmat suunniteltiin ja koodattiin paperilla. Samoin ohjelmia käännettiin konekieleksi paperilla, minkä jälkeen ne syötettiin koneeseen niillä vivuilla. Ohjelmia voitiin toki myös siirtää koneluettavaan muotoon, mutta siihenkin käytettiin usein paperisia tai pahvisia reikänauhoja ja -kortteja, joita rei'itettiin jopa täysin manuaalisesti. Datahaut saattoivat puolestaan sisältää vaiheen, jossa ihmisoperaattori kävi hakemassa reikäkorttipinon arkistokaapista tai kenties motorisoidusta laatikostosta, jossa korteista pidettiin useita kopioita hakua nopeuttamaan (*tub file*). Lopuksi tietokone tulosti ohjelma-ajon tulokset paperille taikka rei'itti ne paperinauhoille tai -korteille. Koneet myös lajittelivat reikäkortteja.

Ensiaskeleet nykyisen kaltaisten näyttöjen ja tekstikäyttöliittymien suuntaan otettiin nekin usein paperilla. Paperia käytettiin monien varhaisten tietokoneiden konsoleihin asennetuissa interaktiivisissa sähkökirjoituskoneissa, joilla tietokonetta käytettiin jotakuinkin nykyisten tekstikonsolien tapaan. Tästä juontuukin edelleen tietotekniikassa käytössä oleva nimitys *konsoli*, joka viittasi alkujaan kalusteeseen. Jatkolomaketta kului, kun koko tietokoneen käyttöistunto tallentui konsolin taakse paperipinoksi. Paperi toimi sekä näyttönä että tallennusmediana.

Paperin runsas käyttö varhaisessa tietotekniikassa onkin ymmärrettävää, kun huomioidaan käytössä ollut kapasiteetti. Esimerkiksi vuonna 1965 myyntiin tullut IBM 1130 -tietokone, jollainen näkyy oheisessa arkistokuvassa, tarjosi ensimmäisessä mallissaan 8 tai 16 kilotavua muistia, ja kiintolevytilaa oli megatavu tai halvimmillaan ei lainkaan. Tulostimien ja reikäkorttien avulla osa tiedosta voitiin säilyttää fyysisessä muodossa paperilla, jolloin se ei vienyt kallisarvoista sähköistä keskusmuistia tai tallennustilaa.

Paperi oli halpaa ja verraten kompaktia. Toisin kuin sähkömuistia, sitä oli myös paljon. IBM-veteraani **Reynold Johnson** on arvioinut, että 1950-luvun alussa 95 % maailman konemuistista oli reikäkorttiarkistoissa. IBM:n parin viikon korttituotanto olisi ulottunut planeetan ympäri. Reikäkortteja käytettiin jopa maksuvälineenä.

Epilogi

Siirtymä paperilta sähköisille medioille tapahtui hiljalleen viime vuosisadan aikana. Esimerkiksi magneettinauhat ja huonekalun kokoiset kiintolevyt edelsivät levykkeitä vuosikymmenillä, joten erilaisia teknologioita käytettiin rinnakkain. Aina siirtymät eivät sujuneet kivutta. Eräs suomalainen ATK-konkari muisteli hiljattain, kuinka kollega oli sadatellut 70-luvun uutuutta: lerppua (levykettä). Sellainen kun ei mahtunut takataskuun reikäkorttinipun tapaan!

Toisaalta harva meistä nuoremmista voi täysin ymmärtää miltä tuntui, kun matkalla asiakaskäynnille tämä huolella järjestetty reikäkorttipino levisi kadulle. Tietoteknisen horisonttinsa avartamiseksi sitä kannattaa kuitenkin yrittää ymmärtää.

Kirjoitus on jatkoa "Tätini oli assembler" -kolumnille (Skrolli 2020.4). Matkan vanhan tietotekniikan syvään päätyyn voi puolestaan aloittaa Skrolleista 2019.1–2019.4, Joe Watsonin neliosaisesta sarjasta Ensimikro ennen mikropiirejä – Kuinka rakensin tietokoneen 1960-luvulla. Skrollissa 2021.1 voi lisäksi tutustua vanhoihin teollisiin tietokoneisiin Suomen tietokonemuseon syväkatsauksessa, ja Skrollissa 2013.3 näytimme kuinka PDP-1-tietokonetta (1959) ohjelmoidaan emulaattorissa. Kaikki nämä lehdet saat maksuttomina pdf-lehtinä osoitteesta skrolli.fi/numerot. Jos et ole vielä valmis siirtymään automaatioteknisiin lukuvälineisiin, vanhoja irtonumeroita on lisäksi myynnissä paperilla alennushintaan osoitteessa skrolli.fi/kauppa.

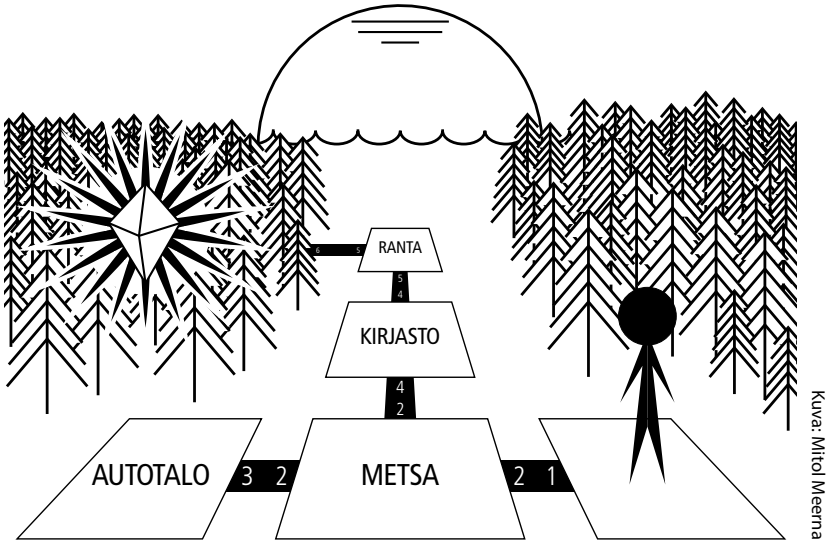
Alakuva: Ei näyttöä tai levykkeitä. Helsingin kauppakorkeakoulun (nykyisin osa Aalto-yliopistoa) ensimmäinen IBM-tietokone otettiin käyttöön vuonna 1968. IBM 1130 -konsolin ääressä on Liisa Korhonen. Kuvan keskellä näkyvät konsolin näppäimistö, tulostin ja merkkivalot sekä vasemmalla lajiteltuja reikäkortteja. Liisan oikealla puolella pöydän alakaapissa on irrotettava kiintolevy. Huomaa "Réfléchissez"-kyltti – kyseessä on IBM:n virallinen plakaatti eli THINK-iskulause ranskaksi. Ajattele! Kuva: Aalto-yliopiston arkisto, Kauppakorkeakoulun kuvalaitos (CC BY 4.0).

TAKA-AJATUS

DATA

Paperille painettu Basic-käsky avasi portin konekieleen ja keinomaailmoihin.

Janne Sirén



Kuva: Mitol Meerna

MikroBitti-lehden numero 11/1987 (s. 59) lupaa vauhtia ja vaaratilanteita sinulle, joka "haluat peittää mielitiettysi timantteihin" ja osallistut *Timanttijahtiin*. "Jahti ei kestä ikuisesti, vaan saattaa loppua hyvinkin lyhyeen, sillä vaaroja tielläsi on lukemattomia." Saatesanoja seuraa tekstiseikkailupelin Basic-kielinen listaus Commodore 16 -kotitietokoneelle. Kuvaus päättyy suoranaisiin ylisanoihin: "Toimitus: Peli herättää huomiota loistavalla grafiikallaan ja sanavalmiudellaan." Kaiken kruunaa **Wallun** piirros kädestä, joka nousee timanttikasan alta.

Lehti tosin lupaa liikoja, sillä pelissä ei ole grafiikkaa, sen "sanavalmis" parseri ymmärtää vain puolen tusinaa käskyä, joista neljä on ilmansuuntia, eikä jahti voi päättyä vaaroihin. Timanttejakin on vain yksi. Timanttijahdissa kiteytyykin lähinnä kaksi kotimikroajan ilmiötä: Ensinnäkin pelin ulkopuolisilla kuvauksilla, kuten pakkauksella tai tässä tapauksessa lehdellä, oli korostunut rooli tunnelman ja kontekstin luojana.

Toisekseen peli on toteutettu data-lauseilla.

Koodattua koodia

Kotimikroaikakaudella 70-luvun lopulta 90-luvun alkupuoliskolle monet kotien tietokoneet, kuten Commodore 16 ja 64, käynnistyivät suoraan Basic-ohjelmointiympäristöön. Niilläkin koneilla, joilla oli jokin muu käyttöliittymä, Basic-kieli sisältyi usein varusohjelmistoon. Näin ollen kotien tietokoneet olivat otollista maaperää Basic-kielisille ohjelmille. Koska tietokoneiden tietoliikenneyhteydet olivat vielä harvinaisia, MikroBitin kaltaiset lehdet olivat keskeinen kanava ohjelmien levityksessä. Basic-ohjelmia levitettiin ajan mittaan myös lehtien oheislevykkeillä ja -kaseteilla, mutta yleisin julkaisutapa 70- ja 80-luvuilla oli ohjelmalistausten painaminen itse lehteen.

Aloittelijoille suunnattu Basic on varsin selkokielinen ohjelmointiväline, kuten monien ensilistaus, tervehdystä ikuisesti toistava esimerkki osoittaa:

```
10 PRINT "MOI"
20 GOTO 10
```

Niinpä lehtien ohjelmalistaukset toimivatkin julkaisukanavan lisäksi sisälukuharjoituksina, joista saattoi poimia ohjelmointioivalluksia omiin projekteihinsa. Aina välillä ohjelmalistausten lukijoita haastoivat kuitenkin kryptiset numero- ja merkkisarjat, joista ei enää ottanutkaan mitään selvää. "Listaus oli data-lauseina."

```
10 FOR I=49152 TO 49361:
READ D: POKE I,D: NEXT
20 DATA 169,0,133,163,1
69,8,133,164,169,80
30 DATA 141,208,192,169
,195,141,209,192,169,0
40 DATA …
```

Data-käskyssä ei sinänsä ole mitään mystistä – sen esiaste löytyy jo 1950-luvulla kehitetystä Fortran-kielestä. Käskyn perässä luetellaan ohjelman tarvitsemia numeroita tai tekstejä, joita luetaan sitten ohjelmakoodin käyttöön read-komennolla. Perus-Basicissa tämä ei kuitenkaan ole kovin intuitiivista, sillä read-komennon lukukohtaa ei määritellä missään ja parametrina annetaan vain kohdemuuttuja. Pitää siis tietää, että read-käsky aloittaa lukemisen listauksen ensimmäisen data-rivin alusta ja askeltaa jokaisella seuraavalla käskytyskerrallaan yhden arvon verran eteenpäin. Lukeminen siirtyy seuraavalle data-riville edellisen päättyessä.

Yllä esitellyssä listauksen pätkässä data-lauseista luetut numeroarvot siirretään poke-komennolla osoitteesta 49152 alkavalle muistialueelle. Tämä oli yksi tapa ladata konekielinen ohjelma tietokoneen muistiin. Data-lauseet sisältävät tällöin kopion konekielellä toteutetun ohjelman muistiesityksestä, minkä vuoksi itse ohjelmalogiikka jää Basic-listauksessa piilon. Listausta ymmärtääkseen se on naputeltava tietokoneelle "sokkona", minkä jälkeen ohjelman voi ajaa muistialueen alkuun osoittavalla `SYS 49152` -komennolla ja konekielistä sisältöä voi tutkia erillisellä konekielimonitorilla, jos osaa.

Mikäli tällaisen konekieliohjelman "Basic-lataajan" toimintalogiikka kiinnostaa, suosittelen COMPUTE!'s Gazette 9/1985 -lehden artikkelia aiheesta. Se löytyy archive.org-verkkokirjastosta (linkki myös Skrollin verkkojatkoilla). Konekielilataajat eivät kuitenkaan olleet ainoa käyttökohde data-lauseille. Dataa käytettiin myös erilaisten vakioiden syöttämiseen Basic-ohjelmille, esimerkiksi pelihahmojen ja pelikenttien määrittämiseen pelimoottoreissa.

Selkokielistä dataa

Yksi selkokielinen käyttötarkoitus data-lauseille olivat tekstiseikkailujen kartat. Data-lauseilla määriteltiin pelialueen paikat ja liikkumismahdollisuudet. Näin tehtiin myös nastolalaisen **Asko Tammelinin** Ti-

manttijahdissa, jonka MikroBitti esitteli edellä mainitussa lehdessään ja jonka äärelle palasimme Askon lähettämän lukijasähköpostin perusteella Skrolli 2021.2:n Postipalstalla.

Asko teki hiljattain Skrollia varten pelille jatko-osan, jonka julkaisemme ohessa. Kiitos, Asko! Se on hyvin uskollinen alkuperäisen pelin rajoituksille. Listauksen rakenne on seuraava:

```
1-9       Alkuruutu
10-2080   Liikkuminen
5000-     Pelilogiikka
```

Rehellisyyden nimissä ensimmäisessä ja viimeisessä osiossa ei ole nähtävää. Pelin ulkoasu ja tekstiä tulkitseva "parseri" ovat MikroBitin (kieli poskella) kirjoittamista ylisanoista huolimatta karuja sekä toiminnaltaan että toteutukseltaan – näin myös uudessa kakkososassa. Karttarivit ohjelman keskellä sisältävät kuitenkin asiallisen esimerkin tekstiseikkailun liikkumismekaniikasta. Tämä mekaniikka perustuu data-lauseisiin.

Listauksessa on myös toinen historiallinen anekdootti. Jokainen rivi päättyy kommenttiin `:REM XX`. Näitä lyhyitä kommentteja ei ole tarkoitettu kirjoitettavaksi ohjelmakoodiin (joskaan ei niistä haittaakaan ole), eikä niillä ole toiminnallista roolia. XX on kunkin rivin `:REM XX` -tekstiä edeltävän osan tarkistussumma. Siitä on hyötyä, jos intoutuu naputtelemaan listauksen autenttisesti käsin ja käyttämään MikroBitti-lehden julkaisemaa C-16-tarkastajaa (esim. MB 6-7/86). Laiskurit voivat hakea *Timanttijahti 2*:n levytiedoston verkkojatkoiltamme (skrolli.fi/numerot). Jahtiin!

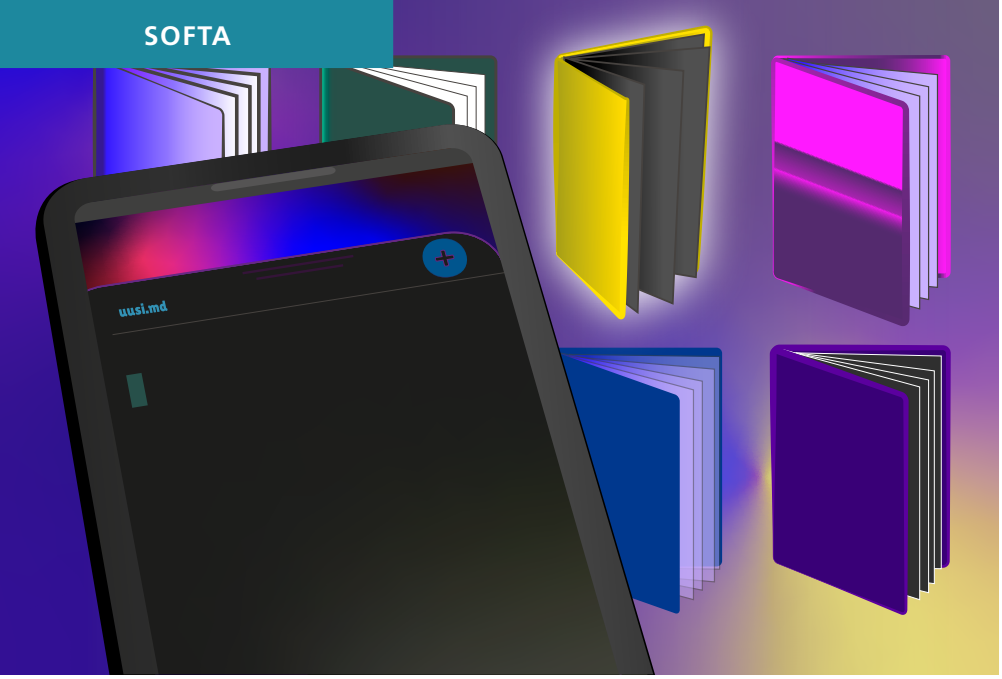
Kirjoitus on jatkoa Bittiseikkailu-kolumnille Skrollissa 2020.2 ja 2021.2:n Postipalstalle (pdf-lehdet maksutta verkkojatkoiltamme). Lisää Tecnopressin tietokonelehdistä s. 88–91.

ASKO TAMMELIN

Timanttijahti 2

C-16

```
1 REM **************************:REM 83
2 REM *  TIMANTIN METSASTYS 2   *:REM CD
3 REM * (C) 2022 JESSE SOFTWARE *:REM AA
4 REM **************************:REM 86
5 COLOR 0,2,0:COLOR 1,2:COLOR
  1,6,4:SCNCLR:REM 91
6 PRINTTAB(11)"TIMANTIN METSASTYS 2":REM
  F4
7 PRINTTAB(11)"START PAINA JOTAIN":REM 4E
8 PRINT TAB(11)"NAPPAINTA":REM 2F
9 GET A$:IF A$<>" " THEN 9:REM D2
10 FOR A = 1 TO 9:REM D3
15 FOR I=1 TO 4:REM DB
20 READ P(A,I):REM D5
26 NEXT:REM A7
30 NEXT:REM A2
31 FOR A = 1 TO 9:READ T$(A):NEXT:REM AF
40 L=2:REM 1F
45 PRINT T$(L):REM 0B
50 INPUT"SUUNTA";S$:REM CB
51 GOTO 5000:REM 64
52 IF S$="P" THEN S=1:REM 2E
54 IF S$="I" THEN S=2:REM 2A
56 IF S$="E" THEN S=3:REM 29
58 IF S$="L" THEN S=4:REM 33
60 IF P(L,S)=0 THEN PRINT"EI KAY":GOTO
  45:REM 1D
70 L=P(L,S):GOTO 45:REM 38
500 REM:REM 79
1000 DATA 0,2,0,0:REM 21
1010 DATA 0,3,4,1:REM 28
1020 DATA 0,0,0,2:REM 23
1030 DATA 2,0,5,0:REM 29
1040 DATA 4,6,0,0:REM 2D
1050 DATA 0,7,0,5:REM 30
1060 DATA 8,0,0,6:REM 33
1070 DATA 0,9,7,0:REM 36
1080 DATA 0,0,0,8:REM 2F
2000 DATA"OLET KOTONA":REM 20
2010 DATA"OLET METSASSA":REM B6
2020 DATA"OLET AUTOTALOLLA":REM 98
2030 DATA"OLET KIRJASTOSSA,NAET
  KIRJAN":REM B8
2040 DATA"OLET RANNALLA":REM A1
2050 DATA"OLET POSTISSA,NAET POMMIN":REM
  F3
2060 DATA"OLET METSATIELLA":REM 8F
2070 DATA"MIES TARJOAA KYYDIN,NAET
  KIRJEEN":REM 8B
2080 DATA"MIES OJENTAA TIMANTIN":REM BC
5000 IF L<>4 THEN 6000:REM 43
5010 IF S$="LUE KIRJA" THEN KA=1:GOTO
  45:REM A9
5020 IF KA=0 THEN GOTO 45:REM 20
5030 S=0:GOTO 52:REM 62
6000 IF L<>6 THEN 7000:REM 47
6020 IF S$="PURA POMMI" THEN MI=1:GOTO
  45:REM 18
6040 IF MI=0 THEN 45:REM F4
6060 S=0:GOTO 52:REM 66
7000 IF L<>8 THEN 8000:REM 4B
7020 IF S$="LUE KIRJE" THEN PI=1:GOTO
  45:REM BD
7040 IF PI=0 THEN 45:REM F8
7060 S=0:GOTO 52:REM 67
8000 IF L<>9 THEN 9000:REM 4E
8020 IF S$="OTA TIMANTTI" THEN EN=1:GOTO
  8060:REM 10
8040 IF EN=0 THEN 45:REM F3
8060 PRINT"PELI LOPPU,HYVIN PELATTU":REM
  32
9000 IF EN=0 GOTO 52:REM F8
9010 PRINT"HALUATKO PELATA UUDELLEEN(K/
  E)":REM 5E
9030 INPUT W$:REM D7
9040 IF W$="K" THEN RUN:REM C7
9050 IF W$<>"E" THEN END:REM E1
```



Paperin jälkeen

Muistiinpanot digiaikana

Ajatuksia ja ideoita syntyy ja katoaa mielessäni niin nopeasti, etten ehdi reagoida niiden sisältöön mitenkään. Muistikirjat ja lehtiöt eivät enää riitä. Tietolaitteille on tarjolla lukematon määrä erilaisia muistiosovelluksia, joista haluan nyt löytää sen oikean.

Teksti ja kuvat: Mitol Meerna

Muistiinpanot ovat konseptina hyvin henkilökohtainen mielen jatke. Muistiin kirjoittamisen ja piirtämisen merkitys on säilynyt myös digiaikana, eikä sen perusidea ole muuttunut. Eri tapoja, tilanteita, tarpeita ja mieltymyksiä muistioiden ominaisuuksien suhteen on suunnilleen yhtä paljon kuin niiden käyttäjiä, ja osittain tämän vuoksi juuri oikeanlaisten välineiden löytyminen voi viedä paljon aikaa.

Kuvittelin kauan sitten, että sopivan yksinkertainen kirjoitussovellus olisi helppoa ja nopeaa löytää: sen kun haen netistä ohjeita ja arvioita eri ohjelmistoista. Kyselin suosituksia ja kokeilin muutamaa. Kyllästyin nopeasti kokeilemiseen, sillä kokeiltavaa oli yhtäkkiä valtavasti, ja sama vaivannäkö toistui aina seuraavan sovelluksen kanssa. Jumituin vuosiksi tilaan, jossa minulla ei ollut yhtäkään kunnollista sovellusta. Muutama kuukausi sitten päätin kokeilla onneani uudelleen.

Arvioin tässä jutussa omien käyttötarkoitusteni ja -tilanteideni perusteella seitsemää muistiosovellusta: Craft, Day One, Drafts, Evernote, Obsidian, Milanote ja IA Writer. Niiden ominaisuuksien lisäksi kerron prosessista, jolla lähdin arvioimaan näitä sovelluksia ja niiden yhteensopivuutta keskenään ja eri pilvipalveluiden kanssa – sekä erityisesti niiden toimintaa kolmen käyttöjärjestelmän eli iOS:n, Windows 10:n ja Linux Mintin, välillä.

Muistiinpanosovelluksista kiinnostuneita saattaa kiinnostaa myös **Janne Sirénin** perehtyminen moderneihin muistiinpanolaitteisiin sivuilla 22–36.

Muistioita metsästän

Tahdon saada aikaan monipuolisen ja kauniin työkalupakin, joka ei vaadi minulta liikaa aktiivista ylläpitämistä ja jonka jokin siihen kuuluva tai kuulumaton palvelu kuljettaa alustalta toiselle enempiä kyselemättä. Tarvitsen muistiota ennen kaikkea kuvittajan, kirjoittajan ja tapahtumajärjestäjän lähtökohdista.

Muistion täytyy sopia juuri minun tarpeisiini, kiinnostuksiini ja toimintatapoihini. Sovelluksia saa olla enemmän kuin yksi, mutta silloin niiden on toimittava hyvin yhteen sekä keskenään että muutaman muun käyttämäni palvelun kanssa. Koska muistion ominaisuuksille ja käytettävyydelle on erilaisia tarpeita eri tilanteissa, on useamman erilaisen löytäminen jopa toivottavaa. Kaipaan yhteensopivuutta muun muassa Google Driveen liittyvien palveluiden kanssa, koska monet projektit, kuten Skrolli, käyttävät sitä yhteistyöskentelyn alustana.

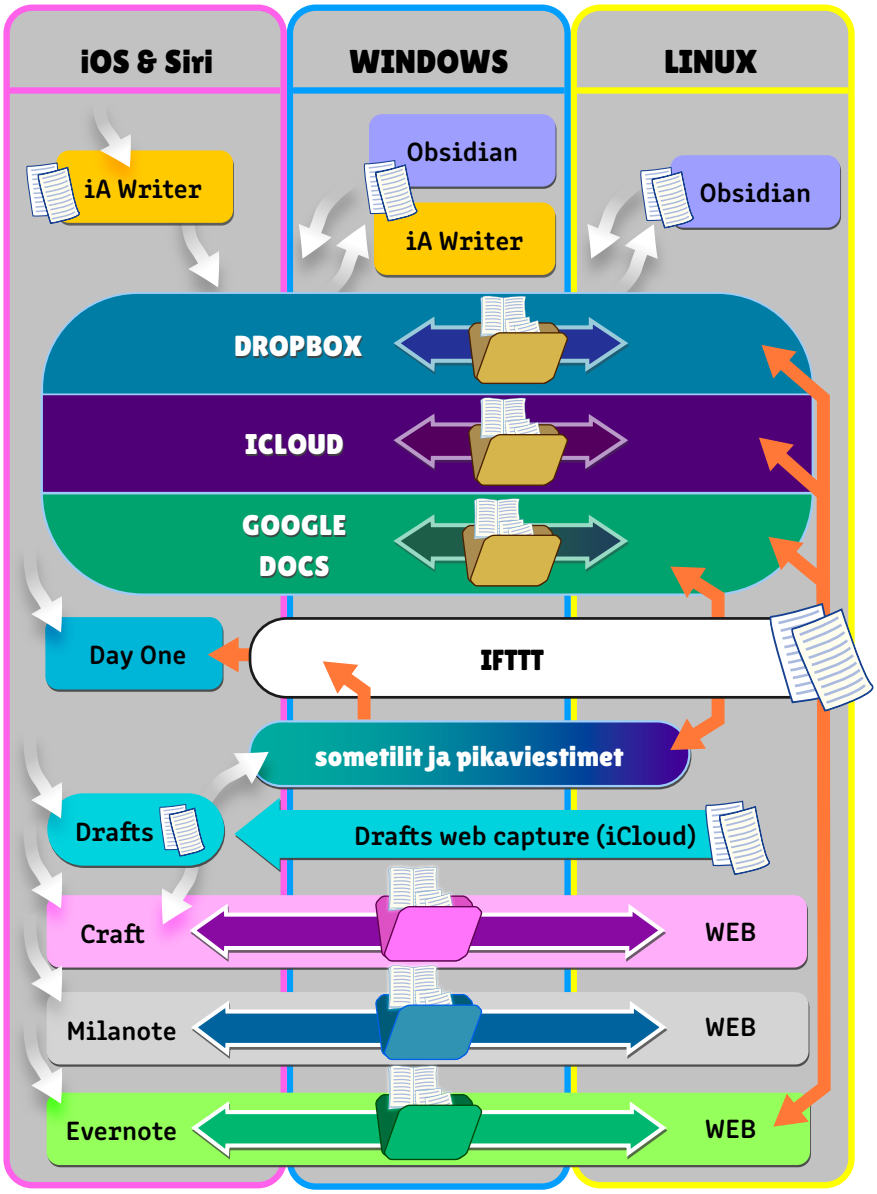
Kirjoittamisen tulee olla mukavaa ensi hetkestä lähtien. Jotta luova työ voi sujua ja onnistua, täytyy siihen käytettävän ohjelmiston olla käyttäjälleen mahdollisimman esteetön. Pieni opettelu alussa ei haittaa, mutta jos käyttöliittymän epämääräisyydet ja tekstin muotoilun vaikeudet hyppivät silmille joka välissä, se saa minut herkästi unohtamaan, mitä olin aikeissa kirjoittaa. Joidenkin sovellusten käytettävyys on niin ongelmallista, että unohdan ylipäätään aikoneeni kirjoittaa jotain. Sovellus toimii intuitiivisesti silloin, kun sen käyttöliittymän voi unohtaa, sen sijaan että unohtaisi, mitä on tekemässä.

Tästä päästään opettelun helppouden tärkeyteen. Sovellusten käyttöohjeiden on oltava selkeällä kielellä kirjoitettuja, seikkaperäisiä ja helposti löydettäviä. Arvostan myös kovasti sellaisia kehittäjiä, jotka ovat tehneet ohjeista tekstin lisäksi videon tai kuunneltavan version. Jotkut eivät kuitenkaan tuota edes alkeellista ohjeistoa perusominaisuuksista. Olen kuluttanut elämästäni lukemattomia tunteja etsien turhaan tietoa eri sovellusten toiminnasta ja yleisistä virheistä. Kun paras arvaus löytyy joltain jo vuosikymmen sitten hylätyltä keskustelualustalta, ollaan todella kaukana käyttökelpoisuudesta.

Jotta kärsivällisyyteni kirjoittamisen, läpiluvun, editoinnin, ja ajatusteni jäsentelyn suhteen ei kärsi, tarvitsen tekstiä rytmittämään visuaalisia apukeinoja eli muotoiluja. Fontin kokoa, tyyliä ja dokumentin värejä on voitava helposti ja nopeasti muuttaa. Myös muut visuaaliset apukeinot, kuten vapaasti sovelluksessa käytettävät ikonit ovat tervetulleita. Dokumentteihin on voitava lisätä kuvia, videon upottaminen on kiva lisä muttei välttämättömyys.

Kaiken tämän pitää tietenkin toimia myös puhelimella kirjoittaessa. Siksi päätin priorisoida muotoilukielenä markdownia, joka on riittävän lyhyttä ja siksi helppo lisättävä tekstiin puhelimellakin – ja samasta syystä se on suhteellisen helppo editoida pois tarvittaessa. Markdown-variaatioita käytetään laajalti nykyaikaisten muistiosovellusten muotoilun pohjana, ja monet muutkin sovellukset tukevat joitain sen peruskomentoja.

Linkin esikatselun esiin saaminen



on minulle tärkeää. Usein pelkkä otsikkokin riittää, mutta kuvan kanssa tunnistan vielä helpommin, mistä on kyse. Ilman esikatseluita joudun keskeyttämään tekemisiäni avatakseni selaimeen toisinaan kymmeniäkin linkkejä kerralla vain nähdäkseni niiden otsikot ja kopioidakseni ne käsin. Siedän tämän tekemistä työpöytälaitteilla tiettyyn rajaan asti, mutta puhelimella työskennellessäni tällainen rasittava ylimääräinen vaiva jää yleensä tekemättä.

Joissain tapauksissa esikatselun puuttuminen johtuu sivuston puutteellisuudesta, yksityisyysasetuksista, maksumuurista tai mahdollisesti jostain muusta välissä olevasta evästeverhosta. Hyvä sovellus osaa silti hakea edes sivuston nimen ja laittaa kuvaksi esimerkiksi jonkin tällaisia tapauksia korvaamaan tarkoitetun grafiikan. Osa sovelluksista antaa muokata sivuston esikatselun muistiossa näkyvää otsikkoa linkkaamisen jälkeen. Tällä tavoin toimivat testaamistani sovelluksista Craft ja Milanote, jotka pohjaavat tiedon visuaaliseen esittämiseen.

Muistiinpanojen säilytyspaikan täytyy olla luotettava datan säilyvyyden, yhtenäisyyden ja koskemattomuuden suhteen. Tarvitsen sellaisen paikan, johon minun on helppo päästä käsiksi, mutta johon muut saavat pääsyn vain silloin, kun haluan.

Pilveen ilman käsiä

Muistiinpanojen yhtenäisyys käyttämiemme laitteiden välillä on elintärkeää kaltaisilleni ihmisille, joilla on paljon mielessä jatkuvasti. Täytyisi siis ehtiä tunnistaa tallennuskelpoiset ajatukset, kääntää ne tunnistettavalle kielelle ja kirjoittaa ylös jonnekin, mistä niitä osaa myöhemmin etsiä ja missä ne ovat turvassa. Minulle ei jää tämän rallin keskellä kaistaa käyttöön miettiäkseni joka välissä, että tallentuiko kaikki ja muistinko painaa nappia.

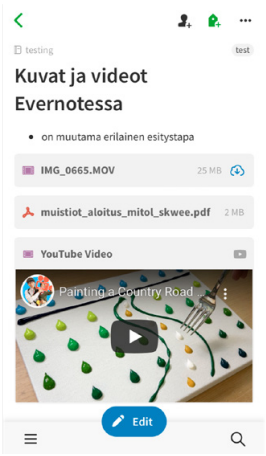
Automaattisen tai ainakin todella helpon ja silti luotettavan tallennuksen mahdollisuuden lisäksi kaipaan muutamia yksinkertaisia tapoja automatisoida muotoiluja ja tiedonkulkua omien laitteideni välillä. Toisinaan haluan voida ohjata puhelintani myös puheella, ja Apple on tehnyt mielestäni hyvää työtä sen mahdollistamisessa laitteillaan. Puheella ohjattavuuden tuki on kuitenkin aina sovelluksesta kiinni. Esimerkiksi muuten monin tavoin loistavasti testeissäni menestynyt Craft ei vielä tue sovelluksen sisällä liikkumista tai ominaisuuksien käyttöä puheella, vaikka siihen voi toki kirjoittaa sanelemalla. Obsidianin mobiilisovellus ei myöskään tue ollenkaan sovelluksen sisällä liikkumista puhuen.

Automaatioiden ja sujuvan tiedonsiirron kannalta Drafts on kirjoitustoimintojensa lisäksi suunniteltu toimimaan eräänlaisena väyläkomponenttina kirjoitusten jakamisessa sovelluksesta toiseen. Sen avulla on mahdollista vähentää tyylimokia vastaanottopäässä, kun halutaan siirtää vaikkapa Microsoft Wordissa luotu teksti kivuttomasti johonkin sovellukseen, joka ei suoraan tue sen oletustiedostoja. Drafts tukee melkoisen pitkää listaa erilaisia tiedostoja, ainakin teoriassa.

Olen yhdistänyt iCloud Driven ja Google Driven toisiinsa siten, että kaikki Applen palvelussa olevat tiedostot ja kuvat siirtyvät myös Googlen puolelle varmuuskopioina. Lisäksi palasin käyttämään Dropboxia äskettäin, sillä huomasin sen kehittyneen palveluntarjoajana mielenkiintoisin tavoin.

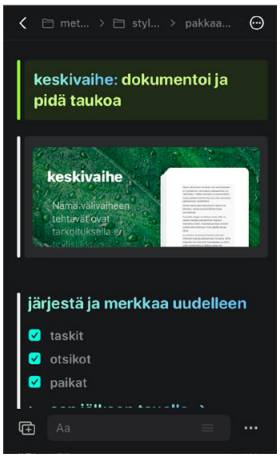
Olen opetellut ohjelmoimaan pieniä mutta käteviä työnkulkuja iPhonen Siri Shortcuts -oikopoluilla. Ne sujuvoittavat puhelimen käyttöä monin tavoin eri sovellusten välillä hyppiessäni. Voin saada esimerkiksi linkeistä tarvitsemani esikatselutiedot näkyviin sellaisiin muistiosovelluksiin, joissa sitä ominaisuutta ei ole tai joissa se ei toimi tarpeeksi luotettavasti. Yksi tekemistäni poluista avaa Dropbox-hakemistoni ja vie sieltä valitsemani tiedostot Craftiin, jos Craft tukee näitä tiedostoja. Tämä on vielä hyvin yksinkertainen mutta nopeuttaa monia tilanteita.

Kaikki iOS:lle tehdyt sovellukset eivät sisällä oikopolkuihin yhdistämisen mahdollisuutta. Joihinkin sovelluksis-



Evernote

Day One



Craft

Drafts

ta, joilla niitä voi käyttää, ovat kehittäjät julkaisseet valmiita pohjia ominaisuuksien hyödyntämiseen. Niiden avulla olen pystynyt oppimaan jonkin verran näiden työnkulkujen syntaksia. Testaamistani sovelluksista Craft, Drafts, Evernote ja Day One on mahdollista yhdistää Sirin työnkulkuihin.

Muista automaatiopalveluista IFTTT on hinnaltaan ja tarjonnaltaan minulle sopiva. Sen käytön opettelu on helppoa ja mielenkiintoista, vaikkakin toisinaan toivoisin enemmän ohjeita automaatioiden luontiin. Onnekseni jotkut IFTTT:hen kytketyt palvelut jakavat itse tietoa siitä, millaisia ominaisuuksia heidän sovelluksestaan voi valjastaa siihen, ja miten niitä kannattaa käyttää. Palvelua voi käyttää hyvin myös maksuttomana, joten jos joskus en enää halua maksaa siitä, en kuitenkaan menetä kaikkia palveluitani ja luomiani yhteyksiä. Ne toimivat hieman hitaammin ja niiden yhteismäärä on rajatumpi.

Evernote

Evernote (evernote.com) oli aikanaan minulle ensimmäinen pilveen perustuva muistiinpanoalusta. Siirryin käyttämään Google Keepiä ja Docsia kymmenisen vuotta sitten. Jokin muisto siitä, että tämä oli joskus kauan sitten mukava muistiinpanoväline, sai palaamaan sen ääreen nyt.

Kaivoin mullasta yhä olemassa olevan tunnukseni ja hämmästelin hetken, miten huonosti olen joskus pitänyt tekstejäni järjestyksessä. Hyvin pian muistin, miksi niin oli. Evernote tarjoaa pitkälti kaikki samat perustyökalut kuin muutkin muistiinpanosovellukset, mutta sen käyttöliittymä on kaukana kannustavasta. Se tuntuu samaan aikaan hyvin kevyeltä ja raskaalta, selkeältä ja sekavalta. On vaikea selittää, mikä tässä niin nyppii. Luultavasti paljon pieniä asioita, jotka kasautuvat nopeasti.

Fonttien määrä on rajattu neljään perusfonttiin, joista yksikään ei miellytä. Linkkien esikatselut, kuvat ja videot toimivat suhteellisen hyvin. Sovellus ei tue markdownia kovin pitkälle itsessään, vaan sitä käyttääkseen sujuvasti Evernoten kanssa täytyy asentaa muiden kehittämä lisäosa, josta ei ole mobiilisovellusta.

Sovelluksen käyttömukavuuden muokattavuus omiin mieltymyksiini sopivaksi on todella hankalaa. Myös maksumuuri tulee vastaan juuri tässä kohtaa: olisi pakko ostaa pro-versio, jos haluaisi voida muokata omaa pääsivuaan niin monta kertaa kuin haluaa. Evernote on minun käyttötapojani ajatellen parhaassa käytössä korkeintaan arkistona sellaisille tiedostoille, joissa tarve aktiiviselle muokkaamiselle on vähäinen. En silti tiedä, miksi ihmeessä käyttäisin tätä mihinkään, mihin Google Drive ja Docs pystyvät - niiden mobiilikäyttökin on monin kerroin mukavampaa kuin Evernoten.

Day One

Aluksi tykästyin Day Oneen (dayoneapp.com) sen yksinkertaisuuden vuoksi, mutta sama ominaisuus sai minut lopulta myös luopumaan sen käytöstä. Yksinkertaisuus itsessään on mukavaa, mutta se on tässä tapauksessa lyöty yli - tai ali, miten sen haluaakin ajatella. Sovellus ei palvele minun käyttötarkoituksiani riittävän hyvin, vaikka siinä on muutamia ihan hyödyllisiä ominaisuuksia.

Day One sopii hyvin sellaisten päivittäisten asioiden keräämiseen, jotka ovat automatisoitavissa mahdollisimman pitkälle ja joihin riittää tarpeeksi simppeli muotoilu. Sillä ei ole mukava kirjoittaa puhelimella, eikä sille ole tällä hetkellä selainsovellusta muille kuin Safarille. Sen käyttö Applen laitteiden ulkopuolella on siis mahdotonta. Sovelluksen kenties kaikkein luotettavimmin toimiva vähän erikoisempi ominaisuus on sen valmiiksi sisäänrakennettu tuki omien Instagram-postauksien peilaamiseen Day Onen pilveen.

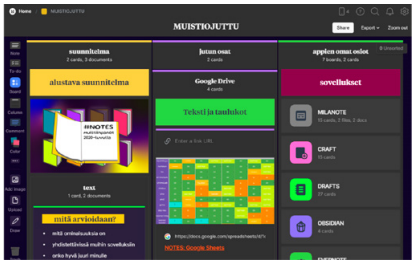
Laitoin Twitteristä ja Facebookista tekemiäni ja seuraamiani somesisältöjä IFTTT:llä kulkeutumaan Day Oneen yksityisiin päivyreihini. Yhteys toimii mutkitta. Somesisältöjen jakamista manuaalisesti Day Oneen sen sijaan haittaa datan ajoittainen putoileminen kärryiltä. Postauksessa oleva kuva tai otsikko saattaa hukkua matkalle, toisinaan taas sovellukseen tulostuu vain tyhjä sivu. Koska Day One ei tue linkin esikatselua, kuvan saaminen linkin mukaan on usein ainoa visuaalinen sisältö mitä postauksiin ylipäätään on mahdollista saada.

Day Onen oma PDF-tuloste on ihan tyylikkäästi toteutettu. Se tarjoaa simppeleitä visuaalisia esityksiä päiväkirjan sisällöstä. Näissä markdownilla tehdyt muotoilut toimivat oikein hyvin. Day One tarjoaa myös ihan printattuja kuvakirjoja omista päiväkirjoista, joita voi tilata sovelluksesta. Voisin esimerkiksi halutessani tilata koko viime vuoden Instagram-postaukseni paperilla kotiin parin kympin hintaan.

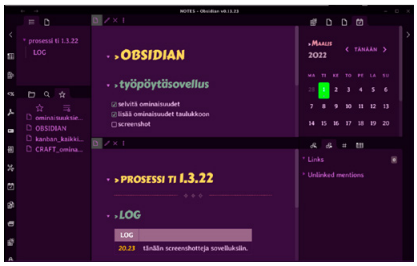
Craft

Craft (craft.do) panostaa tekstin ja sitä ympäröivän mediamateriaalin visuaaliseen esittämiseen mahdollisimman intuitiivisella käyttöliittymällä.

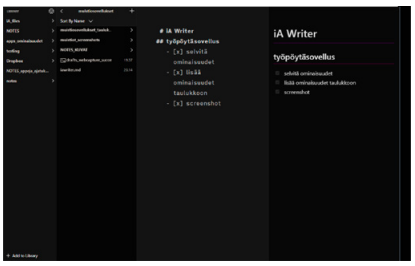
Käytettävyydeltään ja luotettavuudeltaan Craft on tällä hetkellä monipuolisin minun tarkoituksiini. Sovelluksen käytön opin nopeasti erinomaisten oh-



Milanote



Obsidian



IA Writer

jeiden avulla, joihin pääsee käsiksi helposti sekä sovelluksen sisällä että sen verkkosivuilla. Se on juuri sopivan visuaalinen: kuvat, linkkien esikatselut ja haluamani tyylit toimivat hyvin ja niitä on helppo muokata. Craft tarjoaa minulle pitkään kaipaamani kirjoitusmukavuuden mobiililaitteille.

Olen havainnut Craftin varsin toimivaksi työkaluksi ainakin web- ja somesisältöjen kirjoittamisessa, erilaisten hintavertailujen laatimisen apupöytänä sekä opintojen muistioiden tekemiseen. Se tukee markdownin perustyylejä suoraan riviltä ja muuntaa muokattavaan muotoon muista sovelluksista sille syötettyä tekstiä.

Pidempää tekstiä kirjoittaessa ja erityisesti muistiinpanoja kasatessa, editoidessa ja jäsennellessä tulevat Craftin supervoimat esiin. Kirjoittaminen ja editointi sujuvat niin työpöydällä kuin mobiililaitteillakin nopeasti ja luotettavasti ilman turhia sovelluksesta johtuvia keskeytyksiä. On helppoa linkata sivuja toisiinsa ja luoda sisäkkäisiä sivuja, jotka perivät alkuperäisen muokattavuuden. Niitä voi luoda vaikka miten monta sisäkkäin tarpeen mukaan.

Craft on melko uusi tuote markkinoilla ja kehittyy vauhdilla minun kannaltani erittäin hyvään suuntaan. Selainsovelluksen kehitysversio julkaistiin vasta pari kuukautta sitten kiinnostuneille, joten sen keskeneräisyys ei yllätä. Tavoite on kuitenkin julkaista valmis selainsovellus tämän vuoden aikana. Näinä kuukausina, kun olen tehnyt tutkimustani tätä juttua varten, tämä selainversio on ottanut suuria harppauksia käytettävyydessä. Vielä pari viikkoa sitten sillä ei voinut viedä tai tuoda tiedostoja suoraan, vaan tämä piti tehdä puhelimen tai muun Apple-laitteen kautta.

Nyt selaimellakin pystyy tuomaan ja viemään sovellukseen markdown-tiedostoja sekä viemään dokumenteista textbundle-paketteja. Uusin ihastukseni on selaimessakin nyt toimiva, dokumenttiin määrittelemäni tyylit perivä pdf, joka sisältää klikattavina ankkureina kaikki dokumentin sisällä omiksi sivuikseen ryhmittelemäni Craft-dokumentit tyyleineen. Mobiilisovelluksesta saa näiden lisäksi tulostettua suoraan kohdesovelluksessa toimiviksi käännettyjä Microsoft Word -dokumentteja sekä kuvatiedostoja. Docx-tiedostojen jakaminen helpottaa Wordin lisäksi tiedostojen käyttöä Google Docsissa. Tästä huolimatta Craftin mobiilisovelluskin ottaa vastaan vain markdown- ja textbundle-tiedostoja.

Ohjelmistokehittäjille on avattu vastikään omien laajennusten julkaisumahdollisuus ohjeistuksineen. Liityin Craftin yhteisölliseen ja tuki- ja kehitysmieliseen Slackiin, josta olen saanut paljon hyödyllistä tietoa muun muassa mahdollisista työnkuluista, joita Craftin avulla voi suunnitella. Android-sovellusta on lupailtu Craftin sivustolla, mutta sen julkaisulle ei ole vielä annettu minkäänlaista aika-arviota.

Drafts

Draftsin (getdrafts.com) erikoisuus toimia ikään kuin ajatusten portaalina eri appien välillä. Sovelluksen avautuessa se voi näyttää aina tyhjän sivun, jolloin mikään muistiossa jo oleva ei häiritse saapuvaa ajatusta. Sen saa myös avaamaan uuden muistion määritellyn ajan kuluttua. Näen tässä paljon potentiaalia itselleni juurikin tällaiseen käyttöön.

Drafts panostaa erityisesti markdownin tulkkaamiseen mutta antaa mahdollisuuden avata sillä muitakin syntakseja. Ne muuten käyttökelpoiset tekstiä syövät sovellukset, joita ei voi yhdistää suoraan iOS-oikopolkuihin, voisivat hyötyä tästä portaalista. Fontiksi voi valita minkä tahansa järjestelmään asennetuista. Kuvat toimivat vain esikatselussa, joten visuaaliseksi editoriksi tästä ei ole. Niin on toisaalta tarkoituskin, sillä Drafts on alun perin tarkoitettu yksinkertaiseksi ja kevyeksi tekstinkäsittelysovellukseksi.

Draftsissa ei ole perinteistä kansionäkymää, vaan se toimii yksinomaan käyttäjän määrittelemillä filttereillä. Tämä oli itselleni uutta mutta toisaalta mielenkiintoinen ja ajatuksia herättävä lähestymistapa. Päivittäisnäkymäkin on vain filtteri, joka näyttää juuri tänään muokatut tekstit.

Sovellus on julkaistu toistaiseksi vain iOS- ja macOS-laitteille. Selaimella on mahdollista ottaa vaikkapa verkkosivun osoite tai mieleen juolahtanut asia talteen käyttämällä erityisesti tähän tarkoitettua sivua, jolloin se kulkeutuu omaan Draftsiin käytettäväksi myöhemmin. Tämä on helpottava lisä laajempaa selain- tai työpöytäkäytön mahdollisuutta odotellessa.

Draftsin laajempaa käyttöönottoa kohdallani hidastaa se, ettei sovellusta ole vielä julkaistu muille kuin Applen tuotteille, eikä sen käyttö pitkällisempään kirjoittamiseen puhelimella ole riittävän mukavaa. Sen käyttöliittymään on selvästi panostettu, ja sitä voi muokata myös itse monin tavoin, mutta minulle se ei jostain syystä sovi.

Milanote

Milanote (milanote.com) on selainkäyttöön pohjautuva moodboard-tyyppinen visuaalinen suunnittelu- ja julkaisualusta. Sen potentiaali on minun kohdallani suurin visuaalisen suunnittelun työkaluna, johon saa mukaan myös yksinkertaisesti muotoiltua tekstiä. Mobiilisovellus on harmillisen keskeneräinen.

Olen jo pidempään kaivannut jotain Milanoten kaltaista. Sovellukseen saa näkyviin väriarvoja esikatseluineen, kuvia, videoita upotteena ja tiedostona, tekstiä, kommentteja, karttoja, linkkejä esikatselukuvineen sekä ääniupotteita Spotifysta tai Soundcloudista. Näiden formaattien lisäksi siihen voi ladata selainversiossa periaatteessa mitä tahansa tiedostoja, vaikkei Milanote itse niitä pystyisi näyttämään. Ja mikä hauskinta - kaiken päälle voi piirtää. Jos siis haluan tehdä ajatuskartan, Milanote on työpöytäkäytössä todennäköisin työkaluni tähän.

Sovellus tukee muutamia markdownin perustyylejä. Fonttivaihtoehtoja on

vain kaksi. Kirjoittaminen on kuitenkin yllättävän mukavaa näin yksinkertaiseksi muistioksi. Taulukoita Milanoteen ei saa suoraan muokattavaksi, vaan ne täytyy jakaa erillisinä tiedostoina. Se näyttää kuitenkin taulukostakin esikatselukuvan.

Koko sivun voi jakaa valitsemilleen ihmisille tai julkiseksi linkiksi. Tämä on minulle mieleinen ominaisuus, koska se mahdollistaa esimerkiksi ajatuskulkujeni visuaalisen näyttämisen silloin, kun selittäminen ei onnistu.

Mobiilisovelluksella voi tallentaa tekstiä suoraan sovellukseen kirjoittaen, tehtävälistan valintaruutuja ja kuvia. Minkään tiedostojen jakaminen ei onnistu kuvia lukuunottamatta. Nämä rajoitteet muistaen mobiilisovellus tuo kuitenkin Milanoteen alkeellisen puhelimella käytön mahdollisuuden. Kehittäjät ovat luvanneet tähän parannuksia tulevaisuudessa.

Obsidian

Obsidian (obsidian.md) on avoimen lähdekoodin markdown-editori. Se on jo perusteemallaan varsin tyylikäs kokonaisuus, ja muutamalla lisäosalla ja teemaa muovailemalla sain siitä erittäin miellyttävän kirjoitustyökalun, kuitenkin toistaiseksi vain työpöytäkäyttöön. Obsidian Sync on maksullinen synkronointipalvelu, jonka avulla muistiot, tyylit ja asetukset kulkevat saumattomasti koneelta toiselle. Kokeiluni perusteella palvelu pelaa ja on rahan arvoinen, jos ei halua käyttää esimerkiksi Dropboxia, tai jos haluaa muuten vain tukea kehittäjiä. Mobiilisovellus ei valitettavasti ole käyttökelpoinen niiden lisäosien kanssa, mitä tarvitsen kirjoittamisen tueksi, joten jätin sen suosiolla pois opittuani tämän kantapään kautta.

Aivan kaikkea voi muokata. Obsidian sopii siis sellaiseen käyttöön, missä pieni säätö ei haittaa. Säätövaraa on oikeastaan niin paljon, etten nykytiedoillani pysty edes hahmottamaan, mitä kaikkea sille voisi tehdä. Sovellukseen on tarjolla perusteeman lisäksi liuta erilaisiin tilanteisiin, projekteihin ja tunnelmiin sopivia teemoja, joista löysin yhden mukavan myös itselleni. Siitä tuli vielä parempi, kun jaksoin perehtyä sen teemojen toimintaan ja Javascriptiin kiinnitettyihin CSS-tyyleihin. Kuvat ja videot toimivat varsin mukavasti lisäosien avulla.

Taulukoita voi tehdä ja muokata käsin ja siihen tarkoitetun lisäosan avustuksella. Molemmissa tapauksissa taulukoiden muokkaaminen tehdään suoraan markdowniin, jolloin niiden laajempi käyttö muuhun kuin staattisen infon säilyttämiseen ja muuten kuin valmiiden markdown-taulukkopohjien avulla ei ole ainakaan minulle kovin mielekästä.

Sovellukseen saa suosittuna lisäosana tyhjän kalenterin, johon pohjaten voi myös luoda päivittäisiä muistiinpanoja. Kokosin kokeilumielessä pikalisäysskriptin, joka luo kutsuttaessa päivittäisen muistion käyttäen tekemääni taulukkopohjaa. Pohjaan olen muotoillut valmiiksi käskyn tulostaa ensimmäiseen taulukon ruutuun senhetkinen kellonaika, jolloin sitäkään ei tarvitse erikseen lisätä. Se antaa minulle tyhjän kirjoitusruudun ilman hankalia taulukkomuotoiluja, ja näin voin mutkattomasti lisätä kellonajan viereen tekstiä nätisti tulostuvaan taulukkoon. Tämä oli hauska kokeilu, joka auttoi lopulta tämänkin artikkelin kirjoittamista alkuun, sillä se sai minut hahmottamaan aikaa aavistuksen paremmin.

Obsidianin mobiilisovellus ei tue kaikkia lisäosia ja teemoja. Jos siis haluaa käyttää työpöydällä lisäosia, joita mobiilisovellus ei tue, sen käyttöä puhelimella ei voi suositella kahden keskenään synkronoituvan, eri koneilla olevan työpöytäsovelluksen kanssa. Jos taas käyttäisin vain yhtä työpöytäasennusta ja puhelinta, tätä ongelmaa ei todennäköisesti olisi, koska ei olisi tarvetta käyttää ristiriidassa olevia asetuksia. Kokeilin tätä vielä siten, että loin erillisen verkossa olevan kansion eli vaultin puhelinkäyttöön ja käytin sitä työpöydällä ilman mitään lisäosia tai muuta erikoista, mutta sekään ei toiminut halutusti. Syynä on uskoakseni ensinnäkin se, että Applen laitteissa perusfontit ohittavat kaiken muualla sovitun, mikä tulee toisinaan vastaan verkkosivustoja suunnitellessani. Huomasin pian, että mobiilisovelluksella fontin vaihtaminen ei tahtonut onnistua millään. Nämä rajoittuneet fonttiasetukset siirtyivät työpöydälle, ja mikään tyyli ei enää toiminut oikein. ▶

	Craft	Drafts	Evernote	Obsidian	Milanote	iAWriter	Day One
Hinta € / kk Hinta € / vuosi	5 45	2 20	6–9 70–90	9 96	12 120	30 / OS	5 38
Kirjoittaminen	ok	ok	osin	ok	ok	ok	vaikea
Fontit	osin	ok	ei	ok	ei	ok	osin
Kivan näköinen	ok	ok	ei	ok	ok	ok	osin
Tyylien säätö	osin	osin	ei	osin	osin	osin	osin
Kuvat	ok	osin	ok	ok	ok	ok	osin
Video	ok	vaikea	osin	osin	ok	ei	ok
Käytettävyys	ok	ok	vaikea	osin	ok	ok	ok
Markdown	ok	ok	vaikea	ok	osin	ok	osin
iOS-sovellus	ok	ok	ok	vaikea	osin	ok	ok
Siri Shortcuts	ok	ok	ok	ei	ei	ei	ok
IFTTT	ei	ei	ok	ei	ei	ei	ok
Selainkäyttö	ok	ei	ok	ei	ok	ei	ei
Windows	osin	ei	ok	ok	osin	ok	ei
Linux	osin	ei	ei	ok	osin	ei	ei
Pilveen jako	ok	ok	osin	osin	osin	ok	ok
Päivämuistio	ok	osin	ok	ok	ei	ei	ok
Kalenterijako	ok	osin	ok	ei	ei	ei	ok
Taulukot	ok	ok	ok	vaikea	ei	vaikea	vaikea
URL-esikatselu	ok	vaikea	osin	ei	ok	ei	ei
Ääniohjaus	ei	osin	osin	ei	osin	ok	ok

Teksti: Ronja Koistinen
Kuvat: ZA/UM, Wikimedia Commons

KONTSEPTSUALISATSIOON

JA MUUT SANKARIKYTÄN SUPERVÕIMED

Yhden toimittajan rönsyilevä ja vajavainen yritys kertoa jotain siitä, mikä on videopeli Disco Elysium ja keitä ovat sen tekijät. Ei sisällä paljastuksia pelin juonesta, mutta muutamia yksityiskohtia sen maailmasta kuitenkin.

Peruskoulussa minulla oli joukko kavereita, joilla oli menossa ihan oma juttu. He leikkivät koulun pihalla omia leikkejään, kehittelivät omia hahmojaan ja omaa tarinaansa, rakensivat omaa maailmaansa. He piirsivät sarjakuvia ja karttoja maailmansa tapahtumista, kirjoittelivat äidinkielen ja englannin tunneille tarinoita, kuvasivat VHS:lle höpsöjä lyhytelokuvia. Ysärin puolivälissä koodailtiin QBasicillä pieniä maailmaan sijoittuvia pelejä, vuosikymmenen lopulla tehtiin web-sivuja.

Eivät he enimmäkseen olleet erityisen omaperäisiä – vaikutteet *Star Trekistä*, *Tähtien sodasta* ja *Babylon 5:stä* olivat ilmeisiä. Mutta toisaalta mukana oli aina paljon jotain meidän omasta ympäristöstämme, Suomesta ja 1990-luvusta. Suhtautumiseni heihin vaihteli vuosien varrella paljon laidasta laitaan, myötähäpeästä heidän sosiaalisista säännöistä piittaamattomuuttaan kohtaan aina heidän spontaanin tuotteliaisuutensa avoimeen ihailuun. He olivat luovia, ennakkoluulottomia, aikaansaavia ja erityisesti tulenpalavan innokkaita.

Virolainen ZA/UM-kollektiivi on kaiketi vähän samanlainen kaveriporukka. Se julkaisi esikoispelinsä *Disco Elysiumin* vuonna 2019. Peli keräsi oitis valtavasti huomionosoituksia: muun muassa kolme BAFTAa ja D.I.C.E. Awardin sekä useiden medioiden vuoden peli -valinnan. ZA/UM on toki ihan aikuinen ja vakavasti itsensä ottava kulttuurikollektiivi, jonka aiempana julkilausuttuna päämääränä on virolaisen kulttuurin laa-

Karkeasti arvioin, että kyseessä on syvempiä käytettävyysongelmia, joiden ratkomiseen osaamiseni ei tällä hetkellä riitä. Obsidian toimii siten, että kun sen lisäosat, asetukset ja teemat laittaa synkronoitumaan muun datan mukana, myös nämä tiedostot siirtyvät automaattisesti laitteelta toiselle. Tämän myötä myös mobiilisovelluksen rajoitteet siirtyvät työpöydälle, sillä eri laitteilla ei voi laittaa erikseen pois asetusten synkronointia. Näin syntyy tilanteita, joissa sovellus ikään kuin hukkaa asetuksiaan, koska ne eivät voi olla päällä kaikissa käyttökohteissa.

Mobiilisovelluksen käytössä olevat tiedostot tulisi siis saada synkronoitua jotain muuta kautta sellaiseen hakemistoon, johon Obsidianin kooditiedostot työpöytäkäytöstä eivät siirry. Tämä menee sen verran monimutkaiseksi, etten pysty näkemään mobiilikäyttöä kaiken säädön arvoiseksi itselleni. Se pitäisi tehdä aina uudelleen kaikille Obsidianissa käyttämilleni eri päähakemistoille.

IA Writer

IA Writer (ia.net) on markdown-editori, jonka painopiste on kirjoittamisen fokuksessa ja mukavuudessa. Se antaa siistin ja yksinkertaisen käyttöliittymän ajatusten jäsentelyn pohjaksi. Kirjoittaminen tuntuu hyvältä työpöytäsovelluksella ja puhelimella. Niin mukava kuin Writer onkin, yksinään se ei tarpeisiini riitä. Sovellukselle ei ole Linux-versiota lainkaan, eikä se sisällä mitään tiedostojen siirtopalvelua.

Sovellus on ulkomuodoltaan rauhallinen ja elegantti. Writerin markdown-esikatseluun ja editoriin on tarjolla muutama valmis pohja ja fonttivaihtoehto. Näiden lisäksi Writerin kehittäjät tarjoavat mahdollisuuden ja ohjeet pohjien luontiin ja muokkaamiseen. HTML-pohjan ulkoasun muokkaaminen sujuu CSS-tyylien muovailulla, mikä on itselleni tuttua ja mukavaa. Tämä voi kuitenkin nostaa kynnystä pitkällisempään käyttöön niille, jotka haluavat tarjottua enemmän tyylien variaatiota, mutta eivät jaksa samalla tavalla säätää ja opetella koodia.

Sovellukseen saa tuotua markdown-tiedostoja ja Microsoft Word -dokumentteja, ja ulos siitä saa näiden lisäksi pdf- ja html-tiedostoja. Nämä viennit ja tuonnit toimivat moitteettomasti työpöydällä ja puhelimella. Docx-tiedostojen varma tuki antaa jouhevan tavan jakaa Google Drivestä käyttökelpoista tekstiä muotoiluineen Writeriin ja toisin päin.

Writer on testaamistani tekstipohjaisista editoreista selvästi puhelimella käytettävyydeltään paras. Sen rauhallinen ja tuntumaltaan lähes virheetön käyttöliittymä pääsee oikeuksiinsa pienellä kosketusnäytöllä. Muista testin sovelluksista vain Craft yltää käyttömukavuudeltaan samalle tasolle.

Tämän yhtäläisyyden lisäksi havaitsin, että IA Writer ja Craft ovat myös erittäin hyvin yhteensopivia keskenään, sillä kumpikin ottaa toisen tiedostoja vastaan luotettavasti. Tämä yhteensopivuus tukee ja täydentää kummankin sovelluksen käyttökokemusta antamalla luotettavasti toimivan tavan muokata ja pallotella markdownia. Craftin selainsovellus tai Google Docs yhdessä Writer-yhdistettävyyden kanssa mahdollistavat Writerissa luotujen tiedostojen muokkaamisen myös Linuxin puolella.

dun, ryhdin ja omaleimaisuuden vaaliminen.

Tietokoneroolipelien ystävänä yritin pitkään saada pelattua *Disco Elysiumin* loppuun, mutta siinä kesti marraskuuhun 2021. Koin pelin niin temaattisesti raskaaksi ja tunnelmaltaan ahdistavaksi, että jätin sen pitkiksi ajoiksi odottamaan. Tarkoitus oli kirjoittaa siitä jo viime Skrolliin joulukuussa 2021, mutta koska olen joskus vähän *tällainen*, pludasin Elysiumin maailmaan ja ZA/UMin taustoihin sen verran syvälle, että olen vasta nyt valmis sanomaan tästä kaikesta jotakin.

HARMAAN MAAILMA

Disco Elysiumin pelihahmo on ryyppyillan jälkeen muistinsa menettänyt poliisi, joka ei tiedä mitään sen paremmin itsestään, tehtävistään, ajasta, paikasta kuin koko maailmastakaan. Hän on kuin **David Lynchin** ohjaama **Kurt Russell** esittäisi krapulaista Columboa. Kovaksi keitetty, huonoissa elämäntavoissa marinoitu ja menneisyytensä runtelema kyttä on kliseiden klisee, mutta ei se haittaa, jos homman vetää kunnolla. Ja *Disco Elysium* vetää kunnolla.

Tämä on oiva peli *lore*-nörteille. Jos olet sellainen, suosittelen laittamaan pisteitä *Encyclopedia*-kykyyn. Sillä tavalla pelaajahahmo alkaa luetella keskustelujen väliin nippelitietoa valtioista, historiallisista hahmoista, tapahtumista, ryhmittymistä ja sen sellaisesta. Ja sitä tavaraa riittää! ZA/UM on nimittäin työstänyt maailmaansa jo kauan. *Disco Elysium* ei ole edes ensimmäinen kaupallisesti julkaistu teos, joka sijoittuu tähän universumiin. Jo vuonna 2013 ZA/UM kustansi **Robert Kurvitzin** kirjoittaman romaanin *Püha ja õudne lõhn*. Sitä ei ole tietääkseni käännetty (vielä) millekään muulle kielelle, mutta minäpä opiskelin yliopistossa toissa vuosikymmenellä suomalais-ugrilaisia kieliä, enkä ole unohtanut vielä ihan kaikkea viroani, joten ostin kirjan ja luin sen.

Lukukokemus oli haastava. Tarinan kuljetus on hyvin epälineaarista ja poukkoilee ympäriinsä ajassa ja paikassa. Kirja kertoo loppukesällä uimarannalta katoavista opetusministerin tyttäristä ja kaveruksista, jotka 20 vuotta myöhemmin yhä jatkavat näiden etsintää. Henkilöhahmoja on aika paljon, omaperäistä sanastoa vilisee runsaasti ja maailmanrakennus on tiheää.

Kirjan nimi tarkoittaa suomeksi suunnilleen ”Pyhä ja hirvittävä haju”. Englanniksi otsikko on monessa paikkaa käännetty ”Sacred and Terrible Air”, vaikkei kirjaa tosiaan olekaan julkaistu sillä nimellä. Oikeastaan *lõhn* voi tarkoittaa joko hajua tai tuoksua, mutta arvelen, että tässä haju tai jopa löyhkä on miellyttäviä tulkintoja parempi vaihtoehto. Toivon, että kukaan ei kysy, miksi sillä on sellainen nimi.

Kirjassa on jo paikoillaan aika lailla kaikki, mitä pelissäkin. *Iisolat* ja valtiot ovat paikoillaan: Insulinde, Katla, Graad, Samaran kansantasavalta, Vaasa, Revachol. Mainitaan *innossentsit* Dolores Dei, Ambrosius Saint-Miro ja Ernö Pasternak sekä muita historiallisia henkilöitä kuten Kras Mazov. Maailmaa asuttavat oudon nimiset kansakunnat, joilla on jo kirjassa samoja nimiä kuin pelissä: kojkot, meskit, oranjet. Autot eivät ole autoja vaan moottorivaunuja, *mootorkaarikud*, aivan kuten videopelin *motor carriage*. Rahayksikkö on *interisolaarne reaal* ja jopa savukkeiden merkki on Astra. Harmaan läpi matkaillaan erityisillä ilmalaivoilla, joita ei pelissä nähdä, mutta joista voi sieltä täältä lukea.

Niin, se harmaa. Pelatessani peliä ensimmäistä kertaa ryhtini suoristui ja huusin tietokoneen näytölle ”MITÄ HELVETTIÄ”, kun NPC-hahmo kertoi muistinsa menettäneelle etsivälleni harmaasta ja ensimmäistä kertaa valkeni, että Elysium on aivan eri maailma kuin tämä meidän.

Peli kutsuu harmaata nimellä *the Pale*. Kirjassa se on viroksi *hall*, pienellä h:lla, suomeksi kirjaimellisesti harmaa. Kirjassa on sattumoisin myös eräs suomalainen taksikuski, joka kutsuu sitä suomeksi harmaaksi, joten samoin teen minäkin. Tai siis ei suomalainen vaan *surulainen*, joka puhuu *surun kieltä*. Mistäänhän ei Elysiumissa käytetä oikeita nimiä, vaan tosimaailman esikuviin viittaillaan varsin utuisesti ja sürreaalselt. Hulluksi rallikuskiksikin kutsuttu hahmo puhuu myös hyvin kummallisesti. *Suru* on vähän vääristynyttä ja sinne päin kirjoitettua suomea: «Kuuletko sen? Harmaa… se on nyt varmasti harmaa! Mua vähän huolestutta.» Kirjassa vilisee välistä myös puolaa, ehkä tšekkiäkin. Jospa nekin ovat samalla tavalla sürreaalselt kirjoitettuja, enkä vain niitä taitamattomana tiedä?

Harmaa on *TV Tropes* -sivuston termein ”Void Between the Worlds” ja ”Eldritch Location”, tyhjä ja aineeton tila maasaarekkeiden, *iisolien*, välillä. Maailman nimi on Elysium, tai siis *Elyysium* – suomeksi voisi ehkä sanoa Elysion – ja siitä ylivoimaisesti suurin osa on harmaata ei-mitään. Harmaan reunoilla aine hiljalleen murtuu ja liukenee olemattomiin. Harmaata riittää tuhansia kilometrejä suuntaansa, sinne on helppo eksyä ja sen läpi matkustaminen altistaa mielen järkkymiselle – siellä kulkevia tutkimusmatkailijoita kutsutaan entroponauteiksi.

”Kahdenkymmenen kilometrin pääässä etelässä alkaa maailma, Samaran kansantasavalta. Neljäntuhannen kilometrin päässä koillisessa on Katlan iisola, ja sitä, mitä näiden välille jää, ei tiedä kukaan.”

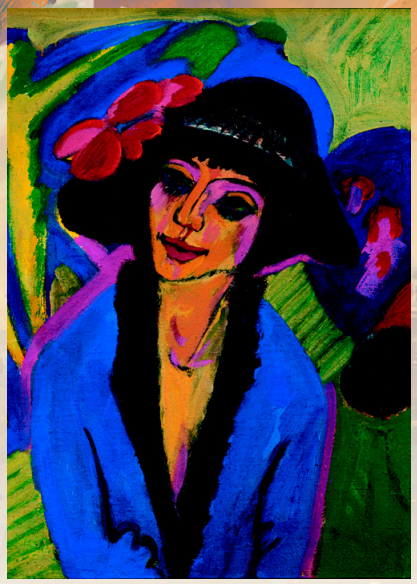
On helppo uppoutua miettimään, mille harmaa voisi olla vertauskuva. Voisiko se olla rautaesirippu, Neuvostoliitto, galaksien välinen tyhjyys, ehkä ilmastonmuutos? Hyvällä symbolilla on tietysti paljon herkullisia tulkintoja, joista valita.

Niin muuten, ZA/UM on kirjassa ihan oma juttunsa. Se on suonensisäisesti annosteltavaa telepaattista lääkeainetta, *kusekollast vedelikku*, kusenkeltaista nestettä. Epäillylle annetaan yksi annos, kuulustelija ottaa toisen, ja sitten kuulusteleva poliisi voi erityisen laitteen avulla nähdä epäillyn mielen sisään.

JÄTTILÄISTEN HARTIAT

ZA/UM-taiteilijat ovat sivistynyttä sakkia. He ovat selvästi lukeneet **Philip K. Dickinsä** ja varmaan myös **Stanisław Leminsä**. Harmaasta ja sen mielenterveysvaikutuksista tulee väistämättä mieleen myös **Stephen Kingin** klassikkonovelli *Jauntti* (*The Jaunt,* 1981), jonka ideana on tähtitieteellisten matkojen kulkeminen silmänräpäyksessä, mutta jos matkustaja ei ole nukutettuna matkan aikana, hänen mielensä tuhoutuu sanoinkuvaamattoman pitkän ajan viettämisestä täydellisessä tyhjyydessä. Samaten mietin suomalaista *Praedor*-roolipelimaailmaa, jossa on niin ikään vaarallisen maailmanrajan takana mystistä taikuutta ja hirviöitä, joita uhmaavat vain koulutetut erikoisjoukot.

Pelin jymymenestyksen myötä käsikirjoittaja **Helen Hindpere** on esiintynyt paljon nettihaastatteluissa, joissa hän on auliisti luetellut ZA/UM-jengin vaikutteita ja kertonut, mitä pelinteon aikana ajateltiin. Erityisesti *Planescape: Torment* (Black Isle Studios, 1999) on mainittu moneen kertaan. Sekä siinä että *Disco*



Vasemmalla Measurehead-videopelihahmo (ZA/UM 2019), oikealla "Gerdan muotokuva" (Ernst Ludwig Kirchner 1914). Ekspressionismia.

Elysiumissa päähenkilö herää muistinsa menettäneenä ja alkaa pikkuhiljaa keräillä maailmansa palasia. Ajattelin pelatessani, että muistinmenetys on aika kulunut klisee, mutta tässä se on tehty oikein tyylikkäästi. Pelaajahahmo paljastuu pian pitkän linjan omapäiseksi koheltajaksi, jolta moista voi odottaakin.

Sitä paitsi muistinsa menettänyt päähahmo on oiva eksposition väline, jonka kautta pelaajan on luontevaa tutustua uuteen maailmaan. Huonosti tehtynä se olisi vaivaannuttavaa kädestä pitämistä, mutta Revacholin kaduilla informaatiohyöky ei ahdista, vaan kukin pelaaja saa uppoutua pelimaailman syövereihin juuri itselleen sopivasti.

Myös graafisesti hurmaavan pelkistetty, minulta aikoinaan kesken jäänyt seikkailupeli *Kentucky Route Zero* (2013) mainitaan vaikuttimena toistuvasti.

VAKAVA TAITEELLINEN NÄKEMYS

Pelin yleisilme on omaperäisesti perinteisen kuvataiteellinen, mistä on vastuussa ZA/UM-studion taiteellinen johtaja **Aleksander Rostov**, öljyväri- ja digitaalimaalari. Hän on pitkäaikainen kollektiivin jäsen, ja jo hänen aiemmassa taiteessaan näkyy selvä uusekspressionistinen tyyli: ronskit siveltimenvedot, uhmakkaat ja kylläiset värit, viitteellisen utuiset taustat, väkevä tunneilmaisu.

Disco Elysiumissa pelihahmolla on 24 eri kykyä, joilla kaikilla on oma muotokuvamaalauksensa hahmolomakkeella. Nämä muotokuvat ovat silkkaa ekspressionismia.

Kykyjen nimet ovat korostetun eriskummallisia ja korkealentoisia. Se on paras osa koko peliä. Jotkin kyvyt ovat tuttuja jokaisesta roolipelistä: Havainnointi (*Perception*), Kestävyys (*Endurance*), Suostuttelu (*Suggestion*). Toiset taas ovat hyvinkin tunnelmallisia ja omintakeisia: Käsitteellistäminen (*Conceptualization*), *Inland Empire* (erilaiset vaistot ja aavistukset), Sähkökemia (*Electrochemistry*), Väristykset (*Shivers*, tietoisuus ympäristöstä), Ryhmähenki (*Esprit de Corps*, kuinka hyvin pelaaja tulee toimeen toisten poliisien kanssa).

Disco Elysiumia pelataan tunteella, irtautuen ja haltioituen kuin näytelmää katsoisi. Tekstiä ja dialogia on paljon – oikeastaan juuri muuta pelissä ei olekaan. Peli on tiivis ja intensiivinen. Miljöötä ja hahmoja on niukasti, pikkukaupunki alkaa tuntua aivan niin ahtaalta kuin se onkin ja pois suljettavien epäiltyjen määrä hupenee äkkiä. Runsas teksti myös vaatii keskittymistä, sillä sanasto on vaativaa ja tyyli anteeksipyytelemättömän väkevää proosaa.

Näytelmäkokemusta tukee myös se, että Revacholin kaupunki on kuin kankaalle maalattu: sitä ei voi hiirellä pyöritellä eri kuvakulmiin, eikä kameraa voi laskea pelihahmon silmien tasolle. Revachol on staattinen kulissi, ja sen tapahtumia seurataan yläviistosta kuin teatterin parvelta.

Erityisen kunnioitettavaa pelissä on, että se ottaa itsensä todella vakavasti vyöryttäessään jopa pateettista melodraamaansa pelaajan silmille – ja lunastaa lupauksensa kirkkaasti, olematta kertaakaan vaivaannuttava tai tahatto-

KÄSITTEELLISTÄMINEN
[Keskitaso: Õnnistui]

Ekspressionismi oli 1900-luvun taitteessa vastareaktio impressionismille. Ekspressionismia ja impressionismia tavallaan yhdistävät muodon viitteellisyys ja painopisteen siirtyminen realistisesta ilmaisusta höttöisempiin vaikutelmiin ja tunteisiin, mutta siinä missä impressionismin jujuna on vangita kuvaan tilan ja hetken vaikutelma (impressio), ekspressionismissa teos on pikemminkin taiteilijan sisäisen tunteen ilmaus (ekspressio).

1800-luvun puolella sellaiset ranskalaiset impressionistit kuin **Claude Monet**, **Édouard Manet** ja **Frédéric Bazille** kuvasivat taiteessaan luontoa, maisemia ja kauniita hetkiä. Impressionistit keskittyivät tilaan, aikaan ja erityisesti valoon. Tilan tuntu, hetken antama vaikutelma (impressio) sekä valo kuvattiin realistisesti ja yksityiskohtaisesti, mutta geometria ja muodon yksityiskohdat jätettiin viitteellisiksi ja suurpiirteisiksi. Ekspressionistisissa kuvissa valon, värin ja tilan realismi ei ole enää mukana, vaan muodot alkavat olla abstraktimpia, kontrastit jyrkempiä, värit ronskeja ja luonnottomia sekä tunnelmat suorastaan vimmaisia.

Pohjoismaista tunnetuin ekspressionistinen maalaus lienee **Edvard Munchin** *Huuto* (1893), vaikka se oikeastaan ehkä edeltääkin ekspressionismia ja on monien mielestä pikemminkin yksi sen esikuvista.

Itse asiassa *zaum* on venäläinen 1900-luvun alun kirjallisuuden, kuvataiteen ja esittävän taiteen tyylisuunta, joka painottaa epäluottamusta sanojen ja kielen merkitykseen sekä korostaa kielen muotoa yli merkityksen. Sana заумь tarkoittaa 'mielen ylittävää; transrationaalisuutta; viisastelua'. Suuntaus oli Venäjällä pinnalla samanaikaisesti kuin ekspressionismi esimerkiksi Saksassa.

ZA/UM

Shivers. Väristysten muotokuva.

Composure. Maltin muotokuva.

man huvittava. Vaikka *Disco Elysium* on erittäin tietoinen edeltäjistään tietokoneroolipelien lajityypissä, se on hyvin vastakohtainen vaikkapa *Falloutin* itsetietoisille *Monty Python* -viittauksille. *Disco Elysium* on kauniin post-ironinen ja vilpitön. Se ei koskaan riko neljättä seinää (lukuun ottamatta harvoja tutoriaalitekstejä), tirsku omille nokkeluuksilleen tai kujeile kieli poskessa.

Pelin tarina ja maailmanrakennus on herkullisen viitteellinen ja arvoituksellinen eikä suinkaan tihku ekspositiota joka rakosesta. Miljöö on rujonkaunis, teknologisesti ja arkkitehtuurisesti jonnekin päin 1900-lukua sijoittuva pieni merenrantakaupunki. Maailmasta välittyy vahva eurooppalaisuuden ja 1900-lukulaisuuden *tunne*, mutta kaikki kansallisuudet, aikamääreet ja tapahtumat ovat vain utuisia viittauksia, eivät mitään todellista.

Fiktiivinen Revacholin kaupunki haisee merituulelta, maaliskuiselta koiranpaskalta, tupakalta ja halvalta aprikoosiesanssilta.

Pelissä on politiikkaa, horjuvia yhteiskuntarakenteita, köyhyyttä, luokkajännitteitä, lohduttomuutta sekä lähimenneisyydestä niskaan huokaileva sota. *Disco Elysiumin* maailmassa on Euroopan merkitykset, värit, äänet ja taakat, joista välittyy vahva sodasta toipuvan Euroopan kokemus, mutta se ei silti ole kirjaimellisesti Eurooppa.

Ainakin Hindpere ja Rostov tiedetään vasemmistolaisiksi, ja arvatenkin muutenkin ZA/UMissa on luettu ylin kyllin vasemmistolaista teoriaa. Siinä määrin syvällisesti *Disco Elysium* nimittäin käsittelee kommunismia ja sen opillisia muotoseikkoja. Pelissä voi löytää pienen kommunistisolun ja ruveta väittelemään siihen kuuluvien nuorten fiksujen miesten kanssa teoriasta.

Kirjassakin puhutaan välistä kommunismista. Tuntuvin lienee kohtaus, jossa entroponautti Zygismunt Berg kiistelee harmaan sisässä kommunismin haamun, *tsütoplasman*, kanssa (suomennos minun):

«Jos niin haluat, niin ole hyvä. Mutta usko mieluummin ihmiseen, älä heihin, usko kommunismiin.»
«Minä jo yritin, mutta en voi! Se ei onnistu minulta… en ole kommunismin tyyppiä.»
«Miksi sitten puhut minun kanssani? Minähän olen kommunismi itse, se haamu, joka kulkee pitkin maata. Miksi olet ollut minun kanssani kaikki nämä vuodet, jos et usko kommunismiin?»
«Vihasta niitä kohtaan, joilla on mennyt elämässä paremmin, Ignus. Ja lisäksi – sinä olet hirviö, groteski. Kukapa ei rakastaisi hirviöiden seuraa?»
«En minä ole hirviö.»
«Sinä olet hirviö, he kutsuvat sinua «apokalyptiseksi teurastajaksi». Keitä muita tunnet, joita kutsutaan sellaiseksi? Ei ketään kutsuta! Kaikki se Graadin verilöyly kävi sinun kättesi kautta, kaikkialla on sinun allekirjoituksesi. Ja vielä silloinkin, kun perääntyessä Mazov ei enää antanut käskyjä, sinä annoit teurastaa vihollisen sotilaat puiden päähän. Kaksitoista tuhatta. Sinä viilasit kuusipuut seipäiksi, teit seivässmetsän, se oli rivoa, Ignus!»
«Se oli siksi, että he antaisivat minun rakentaa oman valtion! Oman tulevaisuuden valtion. Ymmärräthän, he eivät olisi koskaan jättäneet meitä rauhaan… He olisivat ampuneet meidät kuin riistan!»

Kannattaa lukea **Pekka M. Kolehmaisen** artikkeli *Epäonnistumisen poliittinen ideologia Disco Elysium -pelissä* (2020). Kolehmainen kiinnittää huomiota siihen, miten poliittiset ideologiat vahvuuksineen ja varjopuolineen on kudottu taidokkaasti sisään pelin rakenteeseen kaikilla tasoilla. Paitsi että ideologioista puhutaan pelin teksteissä, pelaaja myös osallistuu pelimaailman politiikkaan ja saa mahdollisuuden valita ideologioiden väliltä. Valintojen seuraukset, epävarmuus ja harmaan sävyt rakentavat koherentin kuvan pelimaailman poliittisesta kentästä. ”Artikkeli väittää, että pelimekaanisten ratkaisuidensa avulla peli esittää epäonnistumisen kautta näkemyksiä kuvailemistaan ideologioista ja rakentaa samalla laajempaa maailmankuvaa, jonka keskiössä on epäonnistumisen oma ideologisuus”, sanotaan abstraktissakin.

Aleksander Rostovia on haastateltu vuonna 2010 Tartu Postimees -lehteen. Hän perustelee jutussa taiteilijanimensä valintaa: ”Virallinen nimeni on ikävähkö, mutta Aleksander Rostov on nihilisti, tsaarintappaja, joka heittää dynamiittia vaunujen alle.” Hän viittaa tsaari Aleksanteri II:n murhaan pommiattentaatilla.

JOHTOLANKOJA

Kun hankin Kurvitzin kirjan, pääsin mielenkiintoisille jäljille. Ensinnäkin kirjaan on painettu kaksi nettiosoitetta: sisällysluetteloa edeltävälle lehdelle www.zaum.ee ja takakanteen macaabnyflox.net. Kummassakaan ei ainakaan enää ole mitään nähtävää, enkä ole varma onko jälkimmäisessä koskaan ollutkaan: siitä ei löydy edes historiallista domainin omistustietoa, Internet Archive -vedoksesta puhumattakaan. Kirjan sisäetukannen tyylitellyssä henkilönnimiluettelossa kuitenkin lukee myös MACAAB NYFLOX, eli kyse lienee toistaiseksi esittelemättömästä pelimaailman hahmosta. En muista törmänneeni tähän nimeen kirjassa, eikä Google-hakukaan tuota tuloksia.

Sen sijaan www.zaum.ee on oikein kohtalaisesti arkistoituna Internet Archiven Wayback Machine -palvelussa. Osoitteessa on ollut vuoteen 2019 asti Blogger-alustalla ylläpidetty blogi, joka lienee siivottu pois *Disco Elysiumin* julkaisun aikoihin. Jäljellä on vain 17. syyskuuta 2019 päivätty teksti ”We will be back soon”, joka on paljon mälsempi slõugan kuin vuoden 2014 itsevarma ”Juhtiv ühiskondliku diskussiooni katalüsaator” – johtava yhteiskunnallisen keskustelun muutosvoima. Nykyään ZA/UM majailee osoitteessa zaumstudio.com, joka on tavanomaisen niukka ja sliipattu pelistudion verkkosivu.

Peli sopii oivallisesti meille, jotka emme jaksa aina kiivastahtista toimintakohellusta. Muistinsa menettänyttä juoppodekkaria ohjataan lähinnä verkkaisesti keskustelemalla: jokaista siirtoa saa harkita, eikä koskaan ole kiire.

Blogiin on kirjoiteltu vaihtelevalla aktiivisuudella vuosina 2009–2016. Tekstit käsittelevät kaikenlaista kulttuuriajattelua laidasta laitaan, enkä ole ehtinyt vielä tutustua kaikkeen, vaikka olen koko alkuvuoden aikana käyttänyt ZA/UMin vanhojen internet-jälkien klikkailuun jokusen illan joka viikko.

Esimerkiksi joulukuussa 2012 Robert Kurvitz on kirjoittanut suomalaisista rakkauslauluista. Hän kertoo olleensa vasta sydämensä särkeneenä Suomessa töissä ja kuunnelleensa samalla suomalaista radiota; on vähän hassua lukea näin haltioitunutta ylistystä **Jipusta** ja **Samuli Edelmannista**, mutta mikäs siinä.

”Joka jumalan kappaleessa on suunnitellut sanat, tekstissä on pakko olla ajatusta. Ilman sitä jää kilpailusta ulos. Ja luonnollisesti joka ikinen kappale kertoo rakkaudesta – – ja joka toinen kertoo siitä hyvin. Koska tämä on Suomi ja Suomi on maa, jossa ihmiset yrittävät rakastaa toinen toistaan ja sitten se ei onnistu, ne ovat samalla murheellisia lauluja. – – Millainen maa, sellainen musiikki. – – Siellä on vielä pimeämpi kuin täällä. Vielä kylmempi, kannattaa pitää mielessä. Ehkä meillä ei ole tarpeeksi kylmää ja pimeää ja meidän rakkautemme ovat liian onnistuneita? – –”

ZA/UM-blogi on täynnä Aleksander Rostovin taidetta; Robert Kurvitzin, **Martin Luigan** ja eri vierailijoiden elokuva- ja näytelmäarvosteluja; näyttelyilmoituksia, esseitä, asenteellisia vuodatuksia, valokuvia, runoja ja vaikka mitä; välillä viroksi, välillä englanniksi. Löysinpä ainakin yhden *Pühasta ja õudsesta lõhnasta* poistetun kohtauksenkin.

Vuodelta 2011 löytyy vähän matalaresoluutioinen valokuva käsin piirretyistä luonnoksista otsikolla *Projekt «Vägivald»*: yhdessä paperissa lukee ”Mercutio Dallasz ja kaksteist kojkot”. Tuota nimeä ei esiinny nähdäkseni missään tähän asti julkaistussa Elysium-materiaalissa.

Toinen ZA/UMiin kiinteästi liittyvä sivusto on Nihilist.fm, vuosina 2014–2018 toiminut blogialusta. Sinne kirjoitteli paljon räväkästi virolaisen kirjallisuuden rajoja koetellut ZA/UMin perustaja **Kaur Kender** mutta myös monet muut, esimerkiksi runoilija **Siim Sinamäe**. Nihilist.fm tunnetaan enimmäkseen Kenderin vuonna 2014 blogialustalla julkaisemasta kauhunovellista *Untitled 12*, joka poiki hänelle syytteet lapsipornosta. Novelli on täysin fiktiivinen mutta inhottavan yksityiskohtainen kertomus lapsenraiskaajan elämästä Tallinnassa.

Kender vapautettiin lopulta syytteistä. Tapauksesta kannattaa lukea *Nuori Voima* -lehden juttu vuodelta 2017. Kriittinen ja ulkopuolinen yleisesitys Nihilist.fm:n parhaista paloista taas löytyy *Vikerkaar*-kulttuurilehden nettisivuilta vuodelta 2018. Pääkohdista saanee selvää Google Translaten avulla, vaikkei viroa osaisikaan.

Robert Kurvitz on muuten myös laulaja vuonna 2001 perustetussa Ultramelanhool-progeyhtyeessä, jota on kuvailtu sellaisilla sanoilla kuin ”*pompoosne teatraalsus*” ja ”syntetisaattorin rohkea käyttö”. Biisejä löytyy Youtubesta. Levynkansia on maalannut Aleksander Rostov.

PUHEET JA TEOT

On pelkästään hienoa, että *Disco Elysium* menestyi. Kaikki kiitos on ansaittua, sillä peli on yksiselitteinen mestariteos, jonka pohjatyötä on tehty kauan. Tätä kirjoittaessani on kuulunut epämääräisiä ennakkotietoja siitä, että jatkoa on tiedossa televisiosarjan muodossa. Suunnitelmista ilmoitti Dj2 Entertainment jo vuonna 2020 – sama firma on tuottanut muun muassa *Sonic the Movie* -elokuvan (2020).

Püha ja õudne lõhn oli tarkoitettu kokonaisen scifiromaanisarjan johdannoksi, mutta toista osaa ei ole kuulunut. Itse arvelen osasyyksi Kaur Kenderin oikeudenkäyntiä, jonka jälkimainingeissa hän on sanonut luopuneensa kokonaan kirjojen kirjoittamisesta ja siirtyvänsä käsikirjoittamaan TV-sarjoja ja videopelejä. Hän itse muutti pian oikeusprosessin jälkeen Englantiin, ja koko ZA/UM-studio on mennyt perässä. Kaur Kender oli *Disco Elysiumin* vastaava tuottaja. Hän antoi vuonna 2018 keskeneräisestä peliprojektista Rahageenius-talouslehdelle haastattelun, jossa kertoi itse asiassa, että peliprojekti alkoi jo vuonna 2014 – ehkä saman tien *Untitled 12* -jupakan alettua?

Rahageenius-lehden jutusta: ”Sen maailman luomisessa on Robert Kurvitzilta, **Argo Tuulikilta**, Aleksander Rostovilta ja toisilta mennyt jo pian 18 vuotta. Maailma sai alkunsa kynällä ja paperilla D&D:na, ulos tuli jo romaani *Püha ja õudne lõhn*”, muistuttaa Kender ja lisää, että videopeli oli tähän looginen jatkumo.

Jää nähtäväksi, millainen tulevaisuus Elysionin maailmalla ja siitä vastaavalla ZA/UMilla on. Itse toivon kovasti, että virolainen omaleimaisuus, pystypäinen älykköproosallisuus, kunnianhimoinen taiteellisuus sekä vilpitön poliittinen korkeaotsaisuus säilyvät mukana jatkossakin. Saapa nähdä, mihin suuntaan menestyksen ilmalaiva kuljettaa.

VIITTEET

Helemäe, Deisi 2018. Kaur Kenderi uus videomänguäri on neelanud juba 700 000 eurot investorite raha. *Rahageenius*. raha.geenius.ee/rubriik/uudis/kaur-kenderi-uus-videomanguari-on-neelanud-juba-700-000-eurot-investorite-raha/

Karro, Piret 2018. Laku vittu, Nihilist. *Vikerkaar*. vikerkaar.ee/archives/22517

Kolehmainen, Pekka M. 2020. ”Epäonnistumisen poliittinen ideologia Disco Elysium -pelissä”. *Lähikuva – Audiovisuaalisen Kulttuurin Tieteellinen Julkaisu* 33 (3–4):8–24. doi.org/10.23994/lk.100436

Kunnas, Kaja 2017. Syytetty Kaur Kender. *Nuori Voima* 1/2017. nuorivoima.fi/lue/juttu/syytetty-kaur-kender

Nihilist.fm Internet Archivessa: web.archive.org/web/20181026060651/http://nihilist.fm/

ZA/UM Internet Archivessa: web.archive.org/web/20190703183600/http://www.zaum.ee/

Teksti: Antti Laaksonen
Kuva: Kimmo Rinta-Pollari

Ohjelmointiputka 20 vuotta

Erään koodaussivuston tarina

Vuonna 2001 olin innokas yläastetta käyvä koodari, jonka pääasialliset ohjelmointikielet olivat QBasic ja Visual Basic. Myös nettisivujen tekeminen kiinnosti, ja maailmankuvani mullistui, kun MBnetin kotisivuille tuli mahdollisuus käyttää PHP-kieltä. PHP tuntui poistavan kaikki ikävät rajoitukset, joihin olin törmännyt nettisivujen toteuttamisessa. Kielen suoraviivainen ajattelutapa viehätti myös Basic-ohjelmoijan mieltäni.

Yksi ensimmäisistä PHP-kokeiluistani oli skripti, joka näytti annetun Visual Basic -koodin syntaksiväritettynä HTML-sivulla. Väritys oli yksinkertainen: avainsanat väritettiin siniseksi ja kommentit vihreäksi. Toteutin PHP-skriptin, joka lisäsi koodiin sopiviin väleihin väriä muuttavia font-tageja. Tällaista skriptiä olisi ollut sääli jättää käyttämättä, mistä sain ajatuksen perustaa oman nettisivuston ohjelmoinnin opettamista varten.

Keksin sivustolle nimen Ohjelmointiputka ja avasin sen vuoden 2002 alussa. Valitettavasti en enää muista, miten päädyin nimeen. Tuohon aikaan suomalaista domain-päätettä .fi ei ollut mahdollista varata yksityishenkilönä, ja vaihtoehdot päätteelle olivat käytännössä .com, .net ja .org. Valitsin päätteen .net, jolloin sivuston osoitteeksi tuli ohjelmointiputka.net. Tuntui todella hienolta saada aikaan oma sivusto omalla osoitteella.

Ohjelmointiputkan ensimmäisen version rakentaminen oli minulle PHP-kielen pikakurssi. Toteutin sivustolle keskustelualueen sekä järjestelmän, jonka kautta käyttäjät pystyivät lähettämään koodivinkkejä. Opin kantapään kautta monia tärkeitä asioita nettisivujen ohjelmoinnista. Pikkuhiljaa Ohjelmointiputkaan tuli myös käyttäjiä ja sain heiltä palautetta siitä, miten hyvin tekemäni ominaisuudet toimivat.

Yhteisön muodostuminen

Vuosituhannen alussa netissä oli Ohjelmointiputkan lisäksi monia muitakin suomalaisia sivustoja, joiden tavoitteena oli opettaa ohjelmointia. Ohjelmointiputkan lähin esikuva ja kilpailija oli Mureakuha, joka oli aloittanut toimintansa aiemmin. Myös esimerkiksi Koodaus.net, Suomipelit.com ja Webmaailma käsittelivät osittain samoja aiheita. Miten oma sivustoni voisi erottua muista olemassa olevista?

Minulle tuttu Basic-ohjelmointi soveltui hyvin erottumiseen. Olin huomannut, että Basicin maine muilla suo-

malaisilla sivustoilla ja muuallakin oli huono. Lapsuuteni rakasta ohjelmointikieltä pilkattiin netissä, joten jotain oli tehtävä. Halusin näyttää maailmalle, että Basic on hyvä tapa oppia ohjelmointia, kuten olin tehnyt itsekin. Toisaalta minulla ei ollut paljon vaihtoehtojakaan, koska osasin Basicin lisäksi vain PHP:tä.

Erityisesti minua viehätti Microsoftin vanha ohjelmointiympäristö QBasic. Koska QBasic toimitettiin DOS-käyttöjärjestelmän mukana, monilla oli se koneellaan valmiina ja se oli luonteva tapa ohjelmoinnin aloittamiseen. QBasic oli yksinkertainen ja rajoittunut kieli, jossa oli vanhojen aikojen nostalgiaa. Ryhdyin tehtailemaan Ohjelmointiputkaan QBasic-koodivinkkejä ja ilokseni muut käyttäjät alkoivat toimia samoin.

Ohjelmointiputkan toiminta ei ollut vain omissa käsissäni, vaan kaikki oli kiinni sivuston käyttäjistä. Sivustolle perustettiin IRC-kanava, jonne alkoi muodostua tiivis yhteisö. Suurin osa aktiivisista käyttäjistä oli itseni kaltaisia nuoria ohjelmoijia, joilla oli samanlaisia kiinnostuksen kohteita. Monet olivat peruskoulussa, ja heillä oli runsaasti vapaa-aikaa käytettävissä ohjelmointiharrastukseen. Lähes kaikki olivat poikia.

Muilla sivustoilla Ohjelmointiputkan touhua ihmeteltiin. Eräs Mureakuhan IRC-kanavan keskustelija pohti, miten kukaan voisi ottaa vakavasti sivustoa, jossa 12-vuotiaat idiootit koodaavat QBasicia. Toisaalta Ohjelmointiputkan tarkoituskaan ei ollut olla vakavasti otettava. QBasicin alkeellisuus näyttäytyi meille haasteena samaan tapaan kuin vanhempi sukupolvi hioi taitojaan koodaamalla demoja 1980-luvun tietokoneille.

Oppimista ja esikuvia

Ohjelmointiputkan keskustelualue oli toteutettu alun perin niin, että kaikki viestit tallennettiin palvelimelle samaan tekstitiedostoon. Kun käyttäjä lähetti uuden viestin, se lisättiin tiedoston loppuun. Tämä toimi aluksi hyvin, mutta tiedoston kasvaessa alkoi tulla ongelmia. Tiedoston läpikäynti oli hidasta ja uusia ominaisuuksia oli vaikeaa toteuttaa. Tiedoston sisältö meni myös sekaisin useita kertoja, kun kaksi käyttäjää lähetti viestin samaan aikaan.

Voidakseni toteuttaa keskustelualueen paremmin opettelin käyttämään MySQL-tietokantaa, mikä oli ensimmäinen kosketukseni SQL-kieleen. Sain siirrettyä keskustelualueen sisällön tietokantaan, ja tiedon tallennukseen liittyvät ongelmat ratkesivat. Tämä on yksi esimerkki siitä, miten opettavaista oli ylläpitää Ohjelmointiputkaa. Kun parantelin sivustoa, opettelin uusia tekniikoita sitä mukaa, kun niille oli tarvetta.

Toisella tavalla opettavaista sekä minulle että muille käyttäjille oli tuottaa sisältöä ja osallistua keskusteluihin Ohjelmointiputkassa. Kynnys julkaista oli matala, ja sivustolla oli samanhenkistä porukkaa, jolle oli mukavaa esitellä omia tuotoksia. Usein uuden asian opettelu ja sisällön tuottaminen tapahtuivat samaan aikaan. Esimerkiksi kirjoitin C-ohjelmointia käsittelevää opassarjaa samalla, kun opettelin itse C-kielen perusteita.

Ohjelmointiputkan nuorten aktiivikäyttäjien touhuja seurasi hieman syrjempää joukko vanhempia ohjelmoijia. Heistä muodostui sivustolle jonkinlainen asiantuntijapaneeli, jonka palaute oli usein murskaavaa. Yksi vanhempi ohjelmoija kommentoi, että Ohjelmointiputkan sisältö oli niin huonolaatuista, että kaikkien kannalta paras ratkaisu olisi sulkea sivusto. Oli surullista saada tällaista palautetta omasta sivustosta, mutta toisaalta tämä opetti myös sietämään kritiikkiä.

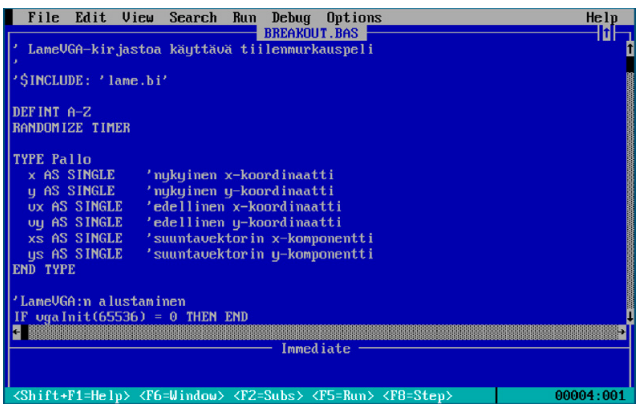
Vanhemmat ohjelmoijat olivat myös tärkeitä esikuvia meille nuorille. Vaikka olimme vielä aloittelijoita, ehkä voisimme joskus oppia yhtä taitaviksi. Tarjolla oli monia suuntia, joihin pystyi syventämään ohjelmointiharrastusta. Tärkein tapahtuma itselleni oli, kun joku keskustelija kertoi käynnissä olevasta Datatähti-kilpailusta. Osallistuin kilpailuun, kiinnostuin sen kautta algoritmeista ja päädyin lopulta yliopistoon tutkimaan algoritmeja.

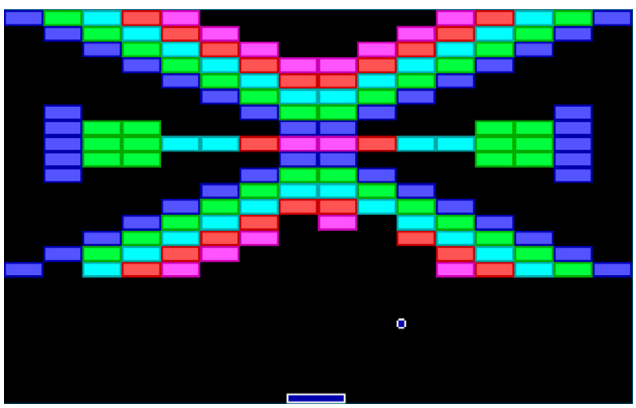
Nuoruus ei ole ikuista

Ohjelmointiputkan alkuvuosina sivustolle tuli jatkuvasti uutta sisältöä ja keskustelua oli paljon. Seurasin sivusta, kun muut suomalaiset ohjelmointisivustot laittoivat lapun luukulle toinen toisensa jälkeen. Syy lopettamiseen oli aina sama: ylläpidolla ei ollut enää kiinnostusta sivuston kehittämiseen. Kun Ohjelmointiputka vietti kymmenvuotisjuhliaan vuonna 2012, muista sivustoista vain Mureakuha sinnitteli enää hengissä ja sulki jonkin ajan kuluttua pysyvästi ovensa.

Ajattelin, että Ohjelmointiputka ei kokisi koskaan muiden sivustojen kohtaloa, vaan jatkaisin uutterasti sivuston parissa tuomiopäivään asti. Toisin kuitenkin kävi, ja aloin käyttää aikaa muihin asioihin. Hylkäsin Ohjelmointiputkan, enkä enää osallistunut aktiivisesti sivuston kehitykseen. Ohjelmointiputkan pääasialliseksi ylläpitäjäksi tuli **Lauri Kenttä**, jonka työn ansiosta sivusto ei ole kuitenkaan vaipunut unholaan.

Vaikka Ohjelmointiputka on edelleen olemassa, jotain on muuttunut. Käyttäjäkunta on vanhentunut samaa tahtia sivuston kanssa, ja nuorten ohjelmoijien yhteisö on kadonnut. Ohjelmointiputkan keskustelualue saattaa olla nyt samassa asemassa kuin Usenetin uutisryhmät vuosituhannen alussa. Kun tutustuin nuorena uutisryhmissä





käytyyn keskusteluun, havaitsin, että vanhat sedät väittelivät siellä menneen maailman asioista. Ohjelmointiputkasta tulee ehkä nyt samanlainen vaikutelma nuorille.

Toisaalta internet on muutenkin muuttunut paljon vuosien kuluessa. Ohjelmointiputkan aloittaessa vuonna 2002 monia nykyään tunnettuja palveluja ei ollut olemassa. Esimerkiksi YouTube, Stack Overflow ja Discord on perustettu vuosia Ohjelmointiputkan jälkeen. Paljon ohjelmointiaiheista sisältöä ja keskustelua on siirtynyt tällaisiin palveluihin. Toiminta tapahtuu kansainvälisessä sosiaalisessa mediassa yksityishenkilöiden sivustojen sijasta.

Kiinnostus oppia ohjelmointia ei ole kuitenkaan vähentynyt Suomessa, vaan pikemminkin päinvastoin. Yliopistojen verkossa järjestämät ohjelmoinnin MOOC-kurssit ovat suosittuja. Ohjelmointi kuuluu nykyään myös koulujen opetussuunnitelmiin. Koodaava nuoriso ei vain enää pyöritä nettisivustoja – mutta missä he ovat ja mitä he tekevät?

Koodia pystymetsästä

Ohjelmointiputkalla on ollut suuri vaikutus uraani. Olen nyt töissä tietojenkäsittelytieteen yliopistonlehtorina Helsingin yliopistossa. Yksi työtehtävistäni on verkkokurssien tuottaminen, jossa teen oikeastaan samaa kuin aikoinaan Ohjelmointiputkassa: rakennan nettisivustoja ja kirjoitan oppimateriaalia. Harrastuksestani on tullut siis ammatti. Ohjelmointiputkan ansiosta opin valtavasti asioita, joista on ollut myöhemmin hyötyä muissa projekteissa.

Teen nykyään yliopistolla töitä myös tietojenkäsittelytieteen tiedekasvatuksen parissa. Yksi tiedekasvatuksen tavoitteista on innostaa nuoria ohjelmoimaan. Minulla on hyvä asema yliopistolla ja mahdollisuuksia osallistua moniin projekteihin. Kuitenkaan mikään yliopiston projekteista ei ole tavoittanut nuoria ja saanut aikaan yhteisöllistä ohjelmoinnin oppimista läheskään yhtä hyvin kuin Ohjelmointiputka.

Miksi pystymetsästä aloitettu Ohjelmointiputka menestyi paremmin kuin yliopiston projektit? Yksi selitys lienee siinä, että Ohjelmointiputka menestyi aitoutensa takia. Tein sivuston omassa elämäntilanteessani itseäni kiinnostavaksi, ja se houkutteli lähes automaattisesti muita samanhenkisiä paikalle. Oppiminen onnistui hyvin, koska olin itse yksi oppijoista ja samalla viivalla muiden kanssa.

Haasteena yliopiston projekteissa on myös, kuinka välttää liiallista tavoitteellisuutta ja passivoivaa ylhäältä päin ohjaamista. Yliopisto tai mikään muukaan taho ei määrittänyt, mitä Ohjelmointiputkassa pitäisi tehdä, vaan me nuoret saimme keksiä sen itse. Oli aikaa tutustua omassa tahdissa asioihin ja myös käydä harhapoluilla. Vaikeutena tiedekasvatuksessa onkin, kuinka edistää oppimista puuttumatta liikaa asioiden kulkuun.

”Sopivan hölmö sopivaan aikaan”

Teksti: Ville-Matias Heikkilä

Irkkilogi paljastaa, että törmäsin Ohjelmointiputkaan ensimmäistä kertaa huhtikuussa 2002. Tällöin se vertautui etenkin Mureakuhaan, joka oli löytynyt saman vuoden helmikuussa. Mieleen tuli myös FW-Sauna-nimisen, suomalaisiin ilmaispeleihin erikoistuneen sivuston ohjelmointifoorumi. Tuntui siltä, että jokainen innokas nuori harrastaja, joka oli saanut webbitilassaan PHP:n ja MySQL:n toimimaan, perusti tuolloin oman ohjelmointifoorumin.

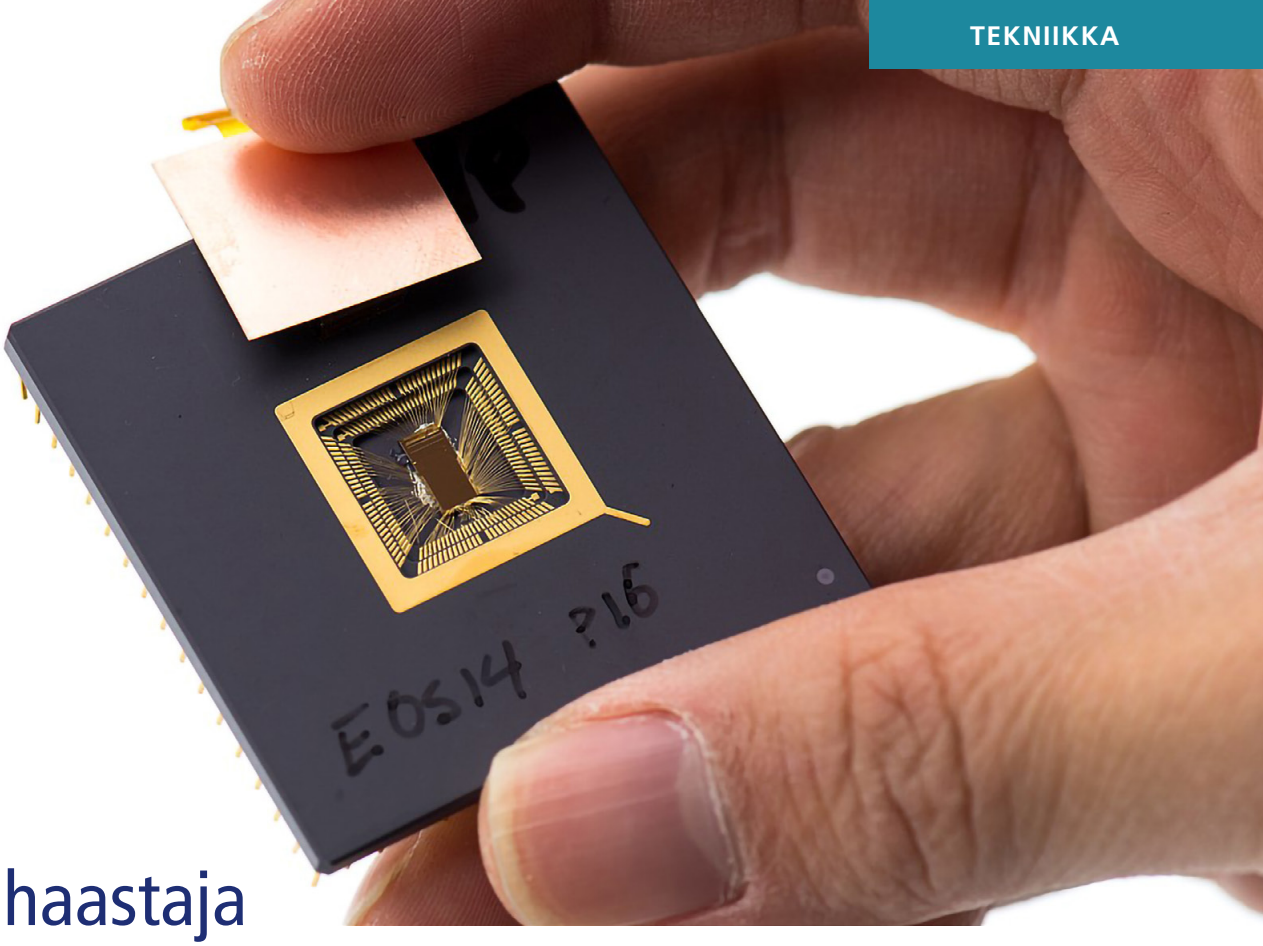
Sivustoissa oli kuitenkin eroja. Kun Mureakuhan innokkaat teinit rakentelivat aikuisten Web 2.0:aa PHP/MySQL:llä, näytti Ohjelmointiputkan puolella Basic olevan kovin sana. Aloittelijat, jotka olivat saaneet yritys-ja-erehdys-menetelmällä aikaiseksi hätäisesti toimivia QBasic-ensiräpellyksiä, jakoivat niitä sivustolla ”ohjelmointivinkkeinä”, joista muut aloittelijat ottivat mallia. En jaa **Edsger Dijkstran** näkemystä siitä, että Basicin opettelu ensimmäisenä kielenä aiheuttaisi pysyvän aivovaurion, mutta sivuston hölmöilykierrettä oli tuskallista seurata. Kuka menisi opastamaan nuo eksyneet lapset oikealle tielle?

Onneksi niin Ohjelmointiputkalla kuin Mureakuhallakin oli oma IRC-kanava. Sivustoja itseään ei kovin isoina annoksina sietänyt, mutta irkissä oli sentään helppo huudella väliin, jos joku neuvoi toiselle täyttä puutaheinää. Joitakin kuukausia kanavilla roikuttuani huomasin, että osa ”putkiksista” oli ruvennut pitämään minua jonkinlaisena puolijumalana tai ainakin ihailun kohteena. Kaikkea sitä joutuikin elämässä kestämään!

Ohjelmointiputkan keski-ikä oli kait hieman matalampi kuin Mureakuhan, tai sitten se oli muuten vain lapsenmielisempi. Kaiken ei tarvinnut olla järkevää, hyödyllistä tai vakavasti otettavaa, vaan asioita saattoi kokeilla jo aivan kokeilemisen ja löytämisen ilosta. Jossain vaiheessa pudotin #mureakuhan pois kanavistani, koska se tuntui niin tylsältä ja toivottomalta tapaukselta, mutta #ohjelmointiputkan ilmapiiri oli hauskempi. Ehkä meininkiin osasi paremmin samaistua omien nuoruudenmuistojen pohjalta.

Näihin aikoihin oli demoskenessä alettu havahtua siihen, ettei uutta verta tullut mukaan niin kuin ennen vanhaan. Löytyisikö Ohjelmointiputkasta kehityskelpoisia yksilöitä? Jos noinkin moni varhaisnuori näki jossain rupuisessa QBasicissa jotain alkuvoimaisempaa ja jalompaa kuin moderneissa webbikikkareissa ja hienouteensa pakahtuvissa GUI-sovelluskehittimissä, oli ainakin mentaliteetti kohdallaan. Moni sitäpaitsi harrasteli varsin lupaavaa grafiikkaohjelmointia, ja olipa muuan 14-vuotias kokeillut ohjelmoida VIC-20-emulaattorillakin, mikä lämmitti sydäntä.

Noh, kyllähän heitä löytyi, ja partyillekin heitä ilmestyi, välillä jopa menestyvien tuotosten kanssa. Myöhemmin en enää onnistunut nuorisorekrytoinnissa lainkaan niin hyvin: kuvalautajonnet olivat jo liian vahvasti eri maailmassa, ja nykyteineistä en edes tiedä, missä he ovat. Mutta kiitos Ohjelmointiputkalle, joka osasi olla sopivan hölmö sopivaan aikaan.



Avoin haastaja

ARM- ja x86-suoritinarkkitehtuurit hallitsevat prosessorien maailmaa, ja vaihtoehdot, kuten MIPS, SPARC ja POWER, ovat jääneet marginaaliin. Näkyvimmäksi haastajaksi valtiaille on noussut RISC-V ("risc five"). Sen päällimmäisenä valttina ARMiin ja x86:een nähden on teknologian avoimuus, mistä seuraa, että kuka tahansa saa tutkia ja muokata sen käskykantaa vapaasti omiin tarpeisiinsa ja valmistaa sen pohjalta prosessoreja. Skrollin lukijoita kiinnostanee myös, mitä tämä RISC-ajattelun juurille palaamista, minimalismia ja modulaarisuutta painottava käskykanta varsinaisesti pitää sisällään.

Teksti: Jari Sihvola
Kuva: Derrick Coetzee / Wikimedia Commons
(Yunsup Lee pitelee RISC-V-prototyyppisirua, 2013)

Kalifornian Berkeleyn yliopiston tutkijat kiinnostuivat avoimeen standardiin perustuvan prosessorikäskykannan kehittämistä, koska he tarvitsivat erilaisiin tutkimustarkoituksiin mukautettuja suorittimia. Intel, AMD ja ARM eivät olleet riittävän kiinnostuneita vastaamaan tällaisiin akateemisiin tarpeisiin, joten luonteva ratkaisu oli lähteä kehittelemään omaa avointa käskykantaa ja suoritinarkkitehtuuria.

Tutkijat eivät ole asialla ensimmäistä kertaa. Arkkitehtuurin nimessä oleva järjestysluku V viittaa professori **David Pattersonin** vetämien suoritinprojektien sukupolviin.

Avoin laajennoksille

RISC-V:n kehitystyö alkoi vuonna 2010 ja julkistamistilaisuus pidettiin 2014 – tärkeimpien käskykantojen suunnittelutyö jäädytettiin laitekehittelyä silmällä pitäen vuonna 2019. Projektin rahoittajaksi löydettiin esimerkiksi Microsoft, DARPA ja ehkä hieman yllättäen myös Intel, joka x86-arkkitehtuuriin tiiviisti kytkeytyvästä historiastaan huolimatta ei ole halunnut rajoittua siihen. Monet isot IT-alan nimet ovat ilmaisseet kiinnostuksensa käyttää RISC-V-prosessoreja tuotteissaan. Nimistä voidaan mainita vaikkapa Samsung, Western Digital, NVIDIA ja Qualcomm. Niin ikään omasta RISC-arkkitehtuuristaan tunnettu MIPS on kiinnostunut RISC-V:stä.

RISC-V:n mahdollisuuksien on huomattu kantavan myös akateemisen maailman ulkopuolelle. Avoimen prosessoriteknologian myötä laitesuunnittelusta on arvioitu tulevan entistä joustavampaa – uudet tuottajat kentällä lisäävät vaihtoehtojen kirjoa, kun aiemmin käskykantojen suunnittelu on jäänyt parin ison nimen tehtäväksi ja laitevalmistajien on pitänyt sopeutua siihen, mitä heillä on tarjolla. Tämän konseptin mukaan erikoistuneisiin tehtäviin, kuten puheentunnistukseen tai kielenkääntöön, voitaisiin laatia omat sirunsa.

RISC-V:ssä tehdään selkeä ero peruskäskykannan, sen virallisten laajennosten ja mukautettujen laajennosten välille. Yhteisten peruslaajennosten päälle voi siis kehittää omia laajennoksia. Tämä lähestymistapa muodostaa selvän eron esimerkiksi ARMiin, jonka suoritinten lisenssiehdot nimenomaan estävät omien laajennosten teon. RISC-V:n kehittäjä Patterson on luonnehtinut, että innovatiivisuus on oleellisesti siirtymässä perinteisestä yhä nopeampien prosessorien kehittämisestä kohti käskykantojen erikoiskäyttöä varten tehtyjä laajennoksia.

On toki esitetty pelkoja, että voimakas kustomointi ja modularisointi voisivat RISC-V:hen pohjautuvien kehitelmien myötä edistää erilaisten sirujen kentän pirstoutumista. Tavoitteena RISC-V:n suunnittelussa on joka tapauksessa ollut ohjelmien yhteensopivuuden säilyminen uudempien, useampia laajennoksia käyttävien prosessoreiden kanssa. Koska laajennosten määritelmät ovat avoimia, olisi lisäksi helpompi tehdä ratkaisuja, joissa tiettyjä käskykannan laajennoksia käyttävä koodi toimii tarvittaessa myös ilman niitä, koska nämä laajennokset voidaan toteuttaa vaihtoehtoisesti ohjelmistokirjastojen kautta.

Käskykannan avoimuus helpottaa luonnollisesti myös teknologian lisensointia. Vertailun vuoksi lisenssien saaminen omisteiseen teknologiaan ei aina ole ollut yksinkertaista; neuvottelut prosessorityyppien käytön lisensseistä ovat voineet olla raskaat ja kestää useamman vuoden. Samoin avoimessa teknologiassa nähdään tietoturvaetuja, koska laitteen voi auditoida murtautumatta siihen ja mahdollisesti päivittää totuttua mutkattomammin.

Teknologian avoimuudella on arvioitu olevan jopa geopoliittista merkitystä, ja jotkut ovat katsoneet tämän näkökannan tulleen yhä painavammaksi koronapandemian aiheuttamien tuotannon häiriöiden myötä. Avoimen teknologian on uskottu olevan paremmin suojassa mahdollisilta muiden hallitusten sekaantumisilta kehitystyöhön samoin kuin paineilta miellyttää osakkeenomistajia. Valtiolliset tahot, mukaan lukien EU, Intia, Venäjä ja Kiina, ovatkin jo osoittaneet kiinnostusta RISC-V:n käyttöä kohtaan omissa strategisissa tietoteknisissä hankkeissaan. Etenkin Kiinalla ja Venäjällä vaikuttaa olevan vahva intressi irtautua riippuvuudestaan läntisiin sirujen valmistajiin.

RISC-V:n tekijänoikeuksia hallinnoi Sveitsissä majaileva RISC-V International -yhdistys (riscv.org), joka myös julkaisee RISC-V:tä määrittävät tekniset dokumentit.

Käskykannan piirteitä

RISC-V:n käskykannan taustalla olevan ajattelun perusidea on pitää käskykanta pienenä ja käskyt yksinkertaisina, kuten alkuperäisessä RISC-paradigmassa (Reduced Instruction Set Computer). RISC-V:n suunnittelijat uskovat, että näin saadaan enemmän irti ennen varsinaista käskyjen ajoa tapahtuvasta käskyjen pakkauksesta (compression) ja pakkaamisen jälkeen tapahtuvasta käskyjen yhdistämisestä (macro-operation fusion). Nämä ovat modernin prosessoriteknologian keinoja tehokkuuden lisäämiseen. Vanhempia arkkitehtuureja suunniteltaessa näitä tekniikoita ei vielä tunnettu hyvin jos lainkaan.

Käskyjen suppeudesta johtuen RISC-V:n konekielinen koodi vaatii usein enemmän käskyjä kuin saman asian tekeminen muilla arkkitehtuureilla. Yksinkertaiset käskyt kuitenkin saadaan pakattua tiheämmin ennen varsinaista suoritusta keskusyksikössä, jolloin niitä voidaan yhdistellä enemmän. Näin lopullinen suoritusvaihe on mahdollisimman tehokas, kun varsinaisiin ajo-operaatioihin saadaan mahdollisimman paljon sisältöä.

Konekielen tasolla vallitseva minimalismi auttaa myös siihen, että suorittimen transistorien määrä saadaan pienemmäksi, jolloin voidaan odottaa tuotantokustannusten madaltuvan ja energiatehokkuuden kohenevan. RISC-V:ssä peruskäskyjen määrä onkin saatu suhteellisen alas, 32-bittisen käskykannan kohdalla 50 käskyyn (40:een, jos aritmeettisten käskyjen immediatemuotoja ei lasketa erikseen). Peruskäskykannan pienessä koossa nähdään ero x86:n ja ARMin käskykantoihin, joista etenkin x86:ta on moitittu paisuneeksi. Laskutavasta riippuen ARMissa on satoja ja x86:ssa tuhansia käskyjä.

Pyrkimys yksinkertaisuuteen näkyy niin ikään RISC-V:n pienessä osoitusmuotojen määrässä. Osoitusmuoto viittaa siihen, miten käskyjen operandit osoitetaan. RISC-V:ssä operandi voi olla rekisterissä tai annettuna välittömästi käskyssä taikka sitten muistiosoitteessa, joka lasketaan joko kaavalla *kantaluku+offset* tai *käskyosoitin+offset*. Tästä saadaan yhteensä neljä osoitusmuotoa (osoitusmuotojen ilmoitettu määrä vaihtelee itse asiassa hieman lähteestä riippuen – asia näyttäisi jättävän hieman tulkinnanvaraa). Myös ARM64:ssä osoitustapoja on pyritty vähentämään ja niitä on listattu olevan viisi. Mitään ARMin tyylisiä hienostuneita silmukoissa käytettäviä laskureiden ja muistiosoitteiden pre- ja post-inkrementointeja ei RISC-V:ssä ole, vaan niiden sijaan inkrementointiin käytetään erikseen tavallista yhteenlaskua. RISC-tyyppisten arkkitehtuurien periaatteiden mukaisesti muistin käyttöön liittyvät lataus- ja säilytyskäskyt ovat tiukasti erillisiä muista käskyistä.

MIPSin tavoin – mutta ARMista ja x86:sta poiketen – RISC-V:n assemblyssä ei ole yhdistetty välittömiä lukuja käyttäviä ja rekistereitä käyttäviä aritmeettisia käskyjä. Monien muiden arkkitehtuurien kohdallahan nämä on abstraktoitu samoiksi käskyiksi. RISC-V:ssä on siis erikseen esimerkiksi käskyt *ADD ja ADDI* eli *add immediate*. RISC-V:n assemblyn *move (MV)* on itse asiassa vain assemblerin tasolla olemassa oleva pseudokäsky, koska rekisterien välinen siirto tapahtuu varsinaisesti *ADDI*-käskyllä. *MV:tä* käyttämällä ei tarvitse kirjoittaa *ADDIn* viimeistä operaattoria, joka olisi aina rekisteristä toiseen dataa kopioitaessa nolla. Kun halutaan laittaa rekisteriin literaaliluku, ei RISC-V:ssä käytetä *movea* vaan *load immediatea (LI)*, joka on niin ikään pseudokäsky.

Yleisrekistereitä RISC-V:ssä on 31 (AMD64:ssä 16, MIPSissä 32, ARM64:ssä 31). Sulautetuille järjestelmille on suunnitteilla myös 16 rekisterin versio. RISC-V:n assemblyssä on MIPSin tapaan annettu selkeyden tavoittelun nimissä päällekkäisiä nimiä yleisrekistereille, joten osaan rekistereistä voidaan viitata esimerkiksi argumenttirekiste-

```
addi   x3, x5, 0     #Data rekisteristä toiseen (x5:stä x3:een) siirtyy
                     yhteenlaskulla,jossa ei lisätä mitään. X3 on
                     kohderekisteri laskun tulokselle.
mv     x3, x5        #Saman voi tehdä pseudokäskyllä ilman nollan
                     kirjoittamista
li     x4, 12        #Load Immediatella laitetaan arvot rekistereihin
                     (12 rekisteriin x4)
```

reinä (*a0-a7*), säilyttävinä rekistereinä (*s0-s11)*, joissa on funktiokutsun ajan säilyvät arvot, ja väliaikaisina rekistereinä (*t0-t6*), joissa on funktioiden käyttämä väliaikainen data. Kutsukonventioiden nojalla paluuosoitteen rekisteri on *x1*, siihen voi viitata myös *ra*:na ja *stack pointer* on *x2* eli *sp*. Lisäksi x3:een voi viitata globaalien muuttujien paikan osoittavana *global pointerina (gp)* ja x4:ään säiekohtaisten yksityismuuttujien paikan osoittavana *thread pointerina (tp)*. Yleisrekisterejä (*x1-x32*) edeltää *x0*, joka on arvoltaan aina nolla.

Itse yritin kirjoittaa ensimmäisen harjoitelmakoodin aluksi pelkkien yleisten rekisterinimien (*x1-x31*) varassa, mutta siirsin dataa heti alkuunsa epähuomiossa paluuosoiterekisterin päälle, minkä huomattuani tuli mieleen, että ehkä vaihtoehtoisten rekisterinimien käyttämisessä voisikin olla ideaa.

Tavujärjestyksen suhteen RISC-V seuraa luonnollisesti suosituimpia nykyajan arkkitehtuureja ja asettaa kahdeksaa bittiä suurempien arvojen tavut järjestykseen alkaen vähiten merkitsevistä tavuista (*little endian,* nouseva tavujärjestys), mutta ARMin tapaan käskyt on suunniteltu niin, että on mahdollista toteuttaa myös muunlaisia tavujärjestyksiä. Näitä ajatellaan tarvittavan lähinnä joustavan yhteensopivuuden saavuttamiseksi vanhoihin laitteisiin, koska nouseva tavujärjestys on muodostunut normiksi.

ARMin kohdalla 32- ja 64-bittiset käskykannat eroavat suuresti toisistaan, sillä ARM64 on julkaistu neljännesvuosisataa alkuperäistä ARMia myöhemmin ja optimoitu uuden ajan tarpeisiin. RISC-V:n kohdalla vastaavaa eroa 32- ja 64-bittisten käskyjen välillä ei ole. Pöydällä on myös 128-bittinen käskykanta, jonka kehitys on vielä kesken. ARMin *Thumb* -käskykannan tapaan RISC-V tarjoaa myös 16-bittiseksi pakatun esitystavan käskyistä (*RV32C*) esimerkiksi mikrokontrollerien käyttöön.

Hallintaa ilman lippuja

RISC-V:ssä on kolme käyttöoikeustasoa: *machine mode, supervisor mode* ja *user mode.* Näistä kullakin on järjestelmän tilan tarkkailuun ja muuttamiseen omat *control status* -rekisterinsä, joita ohjataan omilla pseudokäskyillään, ja kunkin tason toiminnallisuutta voi jatkaa niiden omilla laajennoksilla. Jokainen hardware-säie (RISC-V:n terminologialla ”*hart*”) on aina jollain näistä kolmesta tasosta. Ohjelmat alkavat oletuksena *machine modesta*, jolla on korkeimmat käyttöoikeudet ja näin ollen pääsy järjestelmän matalimpiin tasoihin ja turvallisuuteen liittyviin toimintoihin. *Supervisor mode* liittyy käyttöjärjestelmän toimintaan, *user mode* käyttäjätason ohjelmiin. Yksinkertaisimmat sulautetut järjestelmät saattavat sisältää pelkän *machine moden* tai ainakin jättää pois *supervisor moden*, jota UNIX-tyyppisten käyttöjärjestelmien oletetaan tarvitsevan. Käyttäjätason hardware-säikeet siirtyvät oletustilastaan eli *user modesta* keskeytyskäsittelijän kautta muihin tiloihin, jos esimerkiksi *supervisor*-tilasta tulee kutsu tai aikakeskeytys ilmaantuu.

RISC-V:sta puuttuvat sisäistä tilaa kuvaavat rekisterit, kuten liput, ja operaatioiden tulokset laitetaan siis aina yleisiin rekistereihin. Tämä auttaa toteuttamaan moderneihin prosessoreihin kuuluvaa superskalaariutta, jossa käskyt ajetaan paralleelina vaihtuneessa järjestyksessä, koska ilman statusrekisteriä käskyt pysyvät itsenäisempinä. Vertailuja tekeviä käskyjä, jotka monissa arkkitehtuureissa hyödyntävät lippuja, on RISC-V:ssä yhdistetty haarautuviin käskyihin. Käskyt kuten *BEQ (branch if equal), BNE (branch if not equal), BLT (branch if less than)* ja *BGE (branch if greater than or equal)* ottavat operandeikseen kaksi rekisteriä sekä labelin, johon saatetaan hypätä vertailun tuloksesta riippuen.

Siinä missä ARMin BNE-käsky toimii edellisen CMP-käskyn (*compare*) muokkaaman lipun avulla, RISC-V laittaa operandit samaan käskyyn:

ARM64

```
cmp    x7, x9
bne    loopin_alku
```

RISC-V

```
bne    x7, x9, loopin_alku
```

Toinen esimerkki siitä, miten RISC-V kiertää lippujen käytön, ovat käskyt, kuten SLT (*Set Less Than*), joissa kaksi operandia sisältävät vertailtavat arvot ja tulos (0 tai 1) laitetaan kohderekisteriin:

```
slt    x1, x5, x6
       #jos x5:n arvo on pienempi
       kuin x6:n, x1:een 1
```

RISC-V:n peruskäskykanta koodaa jokaisen käskyn vakiopituiseksi eli 32-bittiseksi ARMin, MIPSin ja POWERin tapaan. Tämä RISC-tyyppisten prosessorien piirre käyttää vakiopituisia käskyjä mutkistaa isojen lukujen käyttöä, koska niitä ei voi ilmaista suoraan yhdellä käskyllä. RISC-V:ssä 32-bittiset luvut ladataan usein kahdella käskyllä, jolloin ensin ladataan LUI:lla (*Load Upper Immediate)* 20 merkitsevintä bittiä ja asetetaan tavallisella yhteenlaskulla loput bitit.

Esimerkkinä 32-bittisen luvun 0xFEF7F6F1 sijoittaminen rekisteriin *a0*

```
lui    a0, 0xFEF7F
       #lataa annetun luvun a0:n
       20 ylimpään bittiin
addi   a0, a0, 0x6F1
       #lisää luvusta loput a0:aan
       ja tulos a0:aan
```

(luvut ovat heksamuodossa, jotta niiden välinen logiikka näkyisi suoraan)

Käskyt, kuten JAL (*jump and link*) ja JALR (*jump and link register*), on tehty tehostamaan funktiokutsujen ja ehtolauseiden käyttöä. Niissä on saman käskyn eri operandeissa rekisteri, johon tulee paluuosoite, sekä osoite (käskystä riippuen label tai rekisteri), johon hypätään.

```
JAL    x1, looppi
       #hyppää labelin ”looppi”
       osoitteeseen ja tallentaa
       paluuosoitteen rekisteriin x1
JALR   x1, 4(x2)
       #sama, mutta hyppää
       osoitteeseen, joka on x2:n
       sisältö plus 4
```

Laajennokset haltuun

RISC-V:n laajentamattoman käskykannan suppeudesta kertoo se, että esimerkiksi niinkin tavallinen asia kuin kokonaislukujen kerto- ja jakolaskut (MUL- ja DIV-käskyt) tulevat mukaan vasta oman laajennoksensa myötä. Liukuluvuille on omat laajennoksensa (erikseen yksin-, kaksin- ja nelinkertaiset tarkkuudet). Järjestelmän hallinnan kannalta tärkeä *control and status* -rekisteri saadaan mukaan laajennoksella, samoin atomiset käskyt, jotka ovat tärkeitä säikeiden ja lukkojen käytössä. Tätä kirjoittaessa myös skalaarisen kryptografian laajennos on jo ratifioitu. Laajennokset käyttäjätason keskeytyksille sekä dynaamisesti tulkattaville

kielille ovat puolestaan kesken. Kaikki edellä mainitut laajennokset ovat ns. peruslaajennoksia (*standard extension*) eli ne ovat yhteensopivia peruskäskykantojen kanssa.

Suurin osa edellä mainituista laajennoksista sisältyisi tavanomaisen pöytätietokoneen prosessoriin. Joissain mikrokontrollereiden prosessoreissa sen sijaan on käytössä pelkät peruskäskyt, esimerkkinä Intelin *Nios V*. Laajentamattomaan käskykantaan sisältyvät vain lataamiset ja säilyttämiset, bitinsiirrot, yhteen- ja vähennyslaskut, loogiset operaatiot (and, or, xor), pienempi kuin -vertailu, hypyt ja haarautumat, funktiokutsu ja ajon lopetus sekä kellon lukemiset.

Peruslaajennoksilla voi vaikuttaa myös säikeiden synkronisointiin. Monien arkkitehtuurien suosima ”vahva muistimalli” rajoittaa prosessoriydinten tuomaa tehohyötyä esimerkiksi synkronoimalla dataa välimuistin kautta. Tässä

```
        addi    sp, sp, -16         #asetetaan pino-osoitin funktion tarpeita varten yhteenlasku-käskyllä
        sd      ra, 8(sp)           #siirretään paluuosoite-rekisteristä arvo pinomuistiin (kohtaan pino-osoitin + 8)
        sd      s0, 0(sp)           #arvon säilyttävistä rekistereistä arvot pinomuistiin. SD = Store Double word
        addi    s0, sp, 16          #pino-osoittimen alkuperäinen arvo talteen kutsun ajaksi (vaikkei sitä tarvita tässä)
        li      a1, 456             #argumentit 123 ja 456 argumentti-rekistereihin a0 ja a1, LI = Load Immediate
        li      a0, 123
        call    testifunktio        #kutsutaan funktiota nimeltä â€œtestifunktioâ€
        li      a5, 0
        mv      a0, a5              #vaikka kutsuttavaa funktiota ei ole edes olemassa, palautetaan silti 0 rekisteriin a0
        ld      ra, 8(sp)           #ladataan paluuosoite pinomuistin kohdasta, johon se aiemmin talletettiin
        ld      s0, 0(sp)           #samoin säilytettävä arvo takaisin (LD = Load Double word)
        addi    sp, sp, 16          #palautetaan pino-osoittimen arvo
        jr      ra                  #hypätään paluuosoite-rekisterin arvoon (Jump Register)
```

Listaus 1. Funktiokutsu RISC-V-assemblyllä (64-bittinen käskykanta) yhden rivin C-koodista GCC:n kääntämänä.

```
        .section .rodata            #osio ohjelman muuttujille, joita ainoastaan luetaan
viesti: .ascii  ”Hei, maailma!\n”   #ASCII-merkkijonomuuttuja ”viesti” sisältöineen.
        .text                       #ohjelman käskyille varattu osio, vain luettava
        .align  1                   #tavuraja seuraavalle käskylle
        .globl  alku                #symboli ”alku” näkyväksi koko ohjelmalle
        .type   alku, @function     #tyyppi symbolille

alku:
        li      a0, 1               #argumentti 1 (file descriptor) argumentti-rekisteriin a0 käskyllä Load Immediate
        la      a1, viesti          #argumentti 2, ladataan viestin muistiosoite käskyllä Load Address
        li      a2, 14              #argumentti 3, viestin pituus tavuina
        li      a7, 64              #Linuxin write-systeemikutsun numero 64 tulee rekisteriin a7
        ecall                       #käsky, joka tekee systeemikutsun a7:n osoittaman numeron perusteella
        li      a7, 93              #lopuksi poistutaan ohjelmasta tekemällä systeemikutsu ”exit”, numero 93
        li      a0, 0               #...johon argumentiksi 0
        ecall
```

Listaus 2. Hei, maailma!” Linuxin write-systeemikutsulla.

```
  char *muuttuja1 = ”esimerkki”;
  char muuttuja2[9];

  for(int i = 0; i != 9; i++) {
    muuttuja2[i] = muuttuja1[i];
  }

        .section .rodata
viesti1:        .ascii  ”esimerkki”
        .data
viesti2:        .byte 9             #varataan 9 tavua
        .text

        la      t3, viesti1         # Load Address, ladataan muuttujien osoitteet rekistereihin
        la      t4, viesti2
        li      t1, 0               # Load Immediate, i = 0
        li      t2, 9               # arvo loopin pysäyttämiseen
loopin_alku:
        lbu     t5, 0(t3)           # Load Byte Unsigned, ladataan tavu t3:n osoitteesta (offset 0)
        sb      t5, 0(t4)           # Store Byte, säilytetään ladattu tavu viesti2:een (offset 0)

        addi    t3, t3, 1           # inkrementoidaan viesti1:n pointteria
        addi    t4, t4, 1           # inkrementoidaan viesti2:n pointteria
        addi    t1, t1, 1           # i++
        blt     t1, t2, loopin alku # jatketaan looppia, jos i < 9, Branch if Less Than
```

Listaus 3. Kopioidaan ASCII-teksti silmukalla toiseen muuttujaan. C-koodi ja sama Assemblyllä

misa	Instruction set architecture, rekisterien leveydet ja käytössä olevat käskylaajennokset
mvendorid	Vendor id, laitteen valmistajan identifioiva JEDEC-arvo
marchid	Architecture id, laitteen arkkitehtuurin identifioiva luku
mimpid	Implementation id, prosessorin tai laitteen mallin tarkemmin identifioiva luku
mstatus	Laitteen status, esim. ovatko keskeytykset ja sivutaulun kääntäminen päällä, virtuaalimuistin mallin valinta
mcause	Tapahtuneiden keskeytysten ja poikkeusten syyt
mtvec	Trap vector handler, trapin käsittelijän muistiosoite, joka laitetaan käskyosoittimeen trapin sattuessa.
mhartid	Hart id, kertoo, mikä prosessoriydin ajaa koodia. Tärkeä, kun tietyn ytimen pitää ajaa tietty koodi.
mepc	Exception program counter, paluuosoite trapista kohtaan, joka aiheutti trapin
mie	Interrupt enable, ajastimen, ohjelmien ja ulkoisten keskeytysten kytkeminen päälle ja pois
mip	Interrupt pending, käsittelyyn tulevat keskeytykset
mtval	Trap value, lisätietoa keskeytyksestä, virheeseen liittyvä osoite tai virheen aikaansaanut käsky
mscratch	Scratch register, tänne voi säilöä muiden käyttöoikeustasojen tietoja väliaikaisesti trapin ajaksi

RISC-V:n laajimmat käyttöoikeudet antavan machine moden omat rekisterit, joiden bitit sisältävät tietoa järjestelmän tarkkailuun ja hallintaan.

mallissa jokainen muutos yhdessä hardware-säikeessä näkyy välittömästi toisessa. RISC-V:n suunnittelussa on pyritty kevyempään muistimalliin, *relaxed memory model*. Tarkoituksena on suoda mahdollisuus välttää synkronointia silloin, kun se on tarpeetonta ja epätehokasta. Käskykannan peruslaajennoksiin sisältyy *FENCE.I* -käsky, jolla voi taata samaan hardware-säikeeseen synkronisoinnin käskymuistiin kirjoittamisen ja käskynhaun välillä. *FENCE.I* oli aluksi osa peruskäskykantaa, mutta se siirrettiin pari vuotta sitten omaksi laajennoksekseen, koska se todettiin vaikeaksi toteuttaa joissakin järjestelmissä ja se osoittautui liian tehottomaksi UNIX-käyttöjärjestelmien käyttäjätasolta käsin käytettynä.

Ehdollisia käskyjä eli käskyjä, jotka suoritetaan vain, jos niihin (niiden operaatiokoodiin) sisältyvä ehto toteutuu, ei RISC-V:n peruskäskykantaan sisälly, mutta ne löytyvät vektorilaajennoksesta. Myös ARM tiputti ehdolliset käskyt 64-bittisestä käskykannastaan. Tämä ratkaisu juontuu siitä, että ehdollisten käskyjen käyttö ei usein ole kovin tarpeellista skalaarimuuttujiin (eli yhden arvon sisältäviin muuttujiin) keskittyvässä koodissa. Näiden käskyjen poissaolon taas katsotaan auttavan superskalaariuden toteuttamista.

Liukuluvut löytyvät RISC-V:n 32 rekisteriä tuovalta laajennokselta. Mukana on käskyjä, joilla voi muuntaa liukulukuja kokonaisluvuiksi ja toisin päin sekä käskyjä, joilla voidaan siirtää dataa kokonaislukujen ja liukulukujen rekistereiden välillä. Tällä vältetään joissakin arkkitehtuureissa oleva pakko siirtää dataa aina muistin lävitse, mistä tulee luonnollisesti tehohävikkiä. Tehokkuutta on pyritty kasvattamaan myös laajennoksen käskyillä, joissa on yhdistetty liukulukujen kerto- ja yhteen- tai vähennyslasku.

Ennen kuin lopetamme pintapuolisen silmäyksemme RISC-V:n Assembly-kieleen ja arkkitehtuuriin, todettakoon, että kirjallisuutta aiheesta näyttää olevan vielä tällä hetkellä varsin niukasti saatavilla, mutta – kenties avoimuuden ja akateemisuuden johdosta – pdf:inä jaossa olevat perusmanuaalit (*The RISC-V Instruction Set Manual I & II*) vaikuttavat sentään suhteellisen selkeiltä verrattuna vaikkapa ARMin 8 000 sivua pitkään referenssimanuaaliin, jota ohjelmointiaiheisilla IRC-kanavilla on neuvottu joskus kiusallaan lukemaan. RISC-V:n manuaaleissa on jopa kerrottu kursiivilla taustoja siitä, miten tiettyihin ratkaisuihin käskykannan suhteen on päädytty.

Tulevaisuus on avoin?

Vaikka edellä on esitetty, että RISC-V on pyritty suunnittelemaan teknisesti edistykselliseksi, sen puolestapuhujat ovat kuitenkin pitäneet nimenomaan avoimuutta RISC-V:n ratkaisevana valttina varsinaisen teknisen laadun sijaan. Nykyään myös SPARC-, POWER- ja MIPS-arkkitehtuurit ovat avoimia, mutta ne eivät ole saaneet samalla tavoin tulta alleen – ehkä lukuun ottamatta IBM:n POWERia, jota on käytetty viime aikoina esimerkiksi muutamissa palvelinkoneissa sekä IBM:n omissa koneoppimisen kehityslaitteissa ja supertietokoneissa. Jotkut ovat nähneet POWERilla olevan vielä mahdollisuuksia kilpailla RISC-V:n kanssa elintilasta.

RISC-V:llä on tosin vielä kovasti matkaa kurottavana laitekehityksen ja ohjelmistoympäristön kehityksen suhteen. Iso joukko sijoittajia ja start-up yrityksiä on kuitenkin jo noussut RISC-V:n junaan. Teknologian aloista on tässä yhteydessä nostettu esiin esimerkiksi esineverkot, koneoppiminen ja autot. Innokkaimmat partisaanit ovat jo tietysti povanneet, että RISC-V tulisi syrjäyttämään x86:n pöytäkoneissa. Kuluttajatason RISC-V-tietokoneita ei kuitenkaan ole vielä näkyvissä, ja myyntiin laitetut Raspberry Pitä muistuttavat laitteetkin ovat kalliita prototyyppejä. Sulautetulla puolella RISC-V on lähtenyt selkeimmin vauhtiin, ja kaikenlaisia RISC-V-arkkitehtuurin prosessorilla varustettuja pieniä kehityslautoja on ollut jaossa jo pidempään. Luonnollisesti RISC-V:tä tuetaan myös monia arkkitehtuureja emuloivassa QEMU-ohjelmistossa (qemu.org).

Avoimuuden suhteen on ehkä hyvä vielä muistuttaa, että vaikka prosessoriteknologia on nyt avointa, on vielä pitkä matka siihen, että muutkin modernin tietokoneen komponentit saataisiin avoimena käyttöön. RISC-V-prosessoreja käyttävät laitteet eivät siis lähtökohtaisesti edusta avointa rautaa kuin rajoitetussa mielessä.

Joka tapauksessa RISC-V-käskykantojen piirustukset on nyt jäädytetty ja hype on puhunut. Seuraavaksi avoin suoritin joutuu tosimaailman testiin.

AJANLASKUJEN ALUT

Skrollissa alkaa uusi juttusarja nimeltä Aikasarja, joka yrittää parantaa lukijoiden ajantajua. Sarjassa tarkastellaan ajan mittaamista, tallentamista, käsittelemistä ja esittämistä tietojärjestelmissä. Sarjan ensimmäisessä osassa tutustutaan eri ajanlaskutapoihin ja niiden nollakohtiin.

Teksti: Ronja Koistinen
Kuvat: Sakari Leppä,
Mitol Meerna

Jo hyvin kauan tietokoneissa ja ohjelmointikielissä on ollut tarpeen käsitellä aikaa. Eri tapoja mallintaa aikaa ja päivämääriä on ollut vuosien saatossa aikamoisesti. Tässä jutussa tutustutaan pintapuolisesti näistä muutamiin.

Pitäydymme nykyajalle mielekkäissä asioissa, emmekä käsittele esimerkiksi vuonna 1975 monenlaiseen sekoiluun johtanutta DECsystem-10-käyttöjärjestelmän päivämäärän ylivuotoa.

Eipä aikailla enempää, vaan mennään asiaan.

1. tammikuuta 1970 – Unix-aika

Nykyajalle ja nykyajan tavallisimmille tietojärjestelmille olennaisin ajanmallinnus on varmaankin Unix-aika. Se on datamalli, jossa aikaa tallennetaan kasvavana kokonaislukuna, jonka yksikkö on sekunti ja nollakohta **1. tammikuuta 1970 klo 00.00 UTC**.

Unix-aikaa käytetään kaikissa Unixin sukuisissa ja POSIX-yhteensopivissa käyttöjärjestelmissä, kuten Linuxissa, BSD:issä ja Androidissa.

Unix-aikaleima 3600 on siis 1. tammikuuta 1970 klo 01.00. Tätä kirjoittaessani Unix-aika on 1643819331, Suomen talviaikaa 2. helmikuuta 2022 klo 18.28.51.

Aikavyöhyke on Unix-ajassa UTC, eli sekuntilukema juoksee kaikkialla maailmassa aina samassa tahdissa. Muunnos paikalliseen aikaan ja mahdolliseen kesäaikaan tehdään aina vasta, kun aikaleima muunnetaan ihmisluettavaan esitysmuotoon. [1]

Nykyään Unix-ajan datatyyppi, `time_t`, on yleensä 64-bittinen etumerkillinen kokonaisluku, mutta vanhastaan se on ollut 32-bittinen. Yhä edelleen 32-bittinen versio on käytössä joissain 32-bittisiä suorittimia käyttävissä sulautetuissa järjestelmissä ja vanhoissa teollisuusjärjestelmissä, joita ei ole syystä tai toisesta päivitetty.

Suurin arvo, joka etumerkillisellä 32-bittisellä kokonaisluvulla voidaan esittää, on 2^{31-1}, siis 2 147 483 647 eli reilut kaksi miljardia. Siitä seuraa, että mittari tulee täyteen klo 3.14.07 tiistaina 19. tammikuuta 2038 (UTC). Tämän ongelman nimi on Y2K38, ja palaamme siihen varmasti Aikasarjassa myöhemmin. Siitä on jo seurannut maailmalla ongelmia, sillä toisinaan tietokoneiden pitää käsitellä myös tulevaisuutta. Esimerkiksi 25 vuoden lainat ovat pankkien järjestelmissä menneet yli tuosta päivämäärästä jo vuonna 2013.

Etumerkillisen kokonaisluvun ylivuoto on C-ohjelmointikielessä *undefined behavior*, periaatteessa siis mitä tahansa voi tapahtua. Käytännössä kaiketi useimmiten käy niin, että arvo kierähtää ympäri maksimiarvosta minimiarvoon: siis tammikuusta 2038 joulukuuhun 1901. Mutta ei ole mitään takeita, että näin tapahtuu kaikissa järjestelmissä.

[1] *Vaikkei Unix-aika välitä aikavyöhykkeistä, karkaussekunnit kuitenkin huomioidaan, koska ne sisällytetään myös UTC-aikaan. Karkaussekunneista ja aikavyöhykkeistä lisää Aikasarjassa myöhemmin!*

Etumerkillisellä 64-bittisellä kokonaisluvulla suurin mahdollinen arvo on 9223372036854775807 eli aika samperistä. Luku on niin suuri, että se ei tunnu kelpaavan Pythonin aikaleimafunktioille. Ilmeisesti tuosta riittää kuitenkin sekunteja sadoiksi miljardeiksi vuosiksi, eli 64 bitillä pärjätään hyvä tovi.

Silti erityisesti sulautetuissa järjestelmissä kuten IoT-laitteissa, ajoneuvoissa, kodinkoneissa, tieteellisissä mittalaitteissa ja teollisuuskoneissa saattaa olla 32-bittisiä Unix-aikatoteutuksia käytössä vielä vuonna 2038. Vaikka kännyköitä ja henkilökohtaisia tietokoneita uusitaankin usein, tällaisten erikoissovellutusten tietokoneet saattavat olla toiminnassa vuosikymmeniä.

Läheisesti Unixiin liittyvässä C-kielessä käytetään `time_t`:n lisäksi myös muutamia muita aikatyyppejä kuten `clock_t` ja `struct timespec`, mutta palataan niihinkin Aikasarjassa joskus myöhemmin.

6. tammikuuta 1980 – GPS

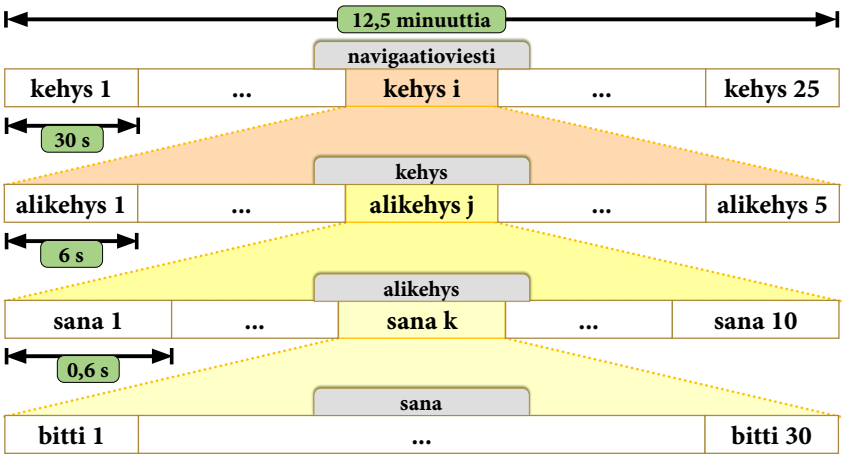
Paikannussatelliittien GPS-järjestelmällä on oma ajanlaskunsa. Satelliittien atomikelloja ei ole synkronoitu UTC-aikaan vaan GPS-järjestelmän omaan aikaan, joka ei muun muassa sisällä karkaussekunteja.

GPS-ajan nollakohta on hieman yllättävä **6. tammikuuta 1980 klo 00.00.00**. Tuolloin GPS-aika ja UTC-aika olivat keskenään synkronissa, mutta karkaussekuntien vuoksi ne ovat sittemmin erkaantuneet toisistaan. Tällä hetkellä GPS-aika on 18 sekuntia UTC-aikaa edellä.

Datamalliltaan GPS-aika on melko erikoinen. Aloitetaan kuitenkin GPS:n navigaatioviestin rakenteesta. Viesti sisältää kolmenlaista tietoa:

- Aikatieto ja satelliitin status
- *Ephemeris*: lähettävän satelliitin tarkkaa kiertoratatietoa
- *Almanac*: kaikkien satelliittien statustietoja sekä karkeampaa kiertoratatietoa

Navigaatioviesti koostuu 30 sekuntia ja 1 500 bittiä pitkistä kehyksistä (*frame*), jotka jakautuvat viiteen kuuden sekunnin mittaiseen alikehykseen (*subframe*). Alikehyksen koko on kymmenen 30-bittistä sanaa. Kukin alikehys sisältää GPS-aikaleiman, joka kasvaa kuudella sekunnilla alikehyksestä toiseen. Alikehys 1 sisältää GPS-viikkonumeron sekä kellonajan korjaukseen ja satelliitin kuntoon liittyvää tietoa. Alikehyksissä 2 ja 3 on *ephemeris* sekä alikehyksissä 4 ja 5 *almanac* (kuva 1).



Kuva 1. L-NAV-tyyppisen GPS-navigaatioviestin lähettämisessä kestää 12,5 minuuttia. Kukin viesti jakautuu 25 kehykseen, joista kukin taas kuuteen alikehykseen. Alikehykset jakautuvat kymmeneen 30 bitin mittaiseen sanaan.

GPS-aika koostuu siis kahdesta osasta: viikkonumerosta ja *time of week* -luvusta (TOW).

Viikkonumero on yksinkertaisesti 10-bittinen kokonaisluku, joka kertoo, montako viikkoa on kulunut sitten 6.1.1980. Amerikkalaiset aloittavat viikkonsa sunnuntaista, joten nollakohtana on sunnuntai 6.1. eikä tiistai 1.1. tai maanantai 7.1. *Time of week* -aika taas on 17-bittinen kokonaisluku, joka on nykyisen viikon alusta kuluneiden kuuden sekunnin mittaisten jaksojen määrän 17 vähiten merkitsevää bittiä.

Satelliitti itse pitää kirjaa 19 bitin tarkkuudella 1,5 sekunnin jaksoissa, mutta HOW-sanaan varattuun 17 bittiin ei mahdu koko viikkoa näin korkealla tarkkuudella. Yksi viikko on 604 800 sekuntia, ja suurin 17 bitillä ilmaistavissa oleva luku on 131 072. TOW saadaan mahtumaan 17 bittiin jakamalla viikko kuuden sekunnin pätkiin: 604 800 ÷ 6 = 100 800. Tästä arvosta päästään myös takaisin alkuperäiseen 19-bittiseen 1,5 sekunnin jaksoissa operoivaan satelliitin sisäiseen esitystapaan yksinkertaisesti kertomalla TOW neljällä.

TOW-aika lähetetään jokaisen alikehyksen alussa *hand-over word* -nimisessä sanassa (HOW). GPS:n tiedonsiirtonopeus on varsin hidas – alikehyksen lähettämiseen kuluu kokonaiset kuusi sekuntia –, joten TOW tarkoittaa itse asiassa seuraavan alikehyksen alkua, eli se on hivenen tulevaisuudessa.

GPS-alikehysten aikaan liittyvistä tietueista tarkemmin seuraavalla sivulla taulukoissa 1 ja 2.

Tässä aikaformaatissa ilmeinen ongelma on tietysti se, että kymmeneen bittiin ei mahdu kuin 1 024 viikkoa, joten viikkonumero pyörähtää ympäri noin 19,7 vuoden välein. Edellisen kerran näin tapahtui huhtikuussa 2019. Tästä seuraa myös se, että GPS-vastaanotinlaitteiden pitää jotenkin ohjelmistossaan varautua siihen, että viikkonumero hyppää joskus tulevaisuudessa taaksepäin.

Yksi tapa on esimerkiksi tallentaa aikaleima muistiin muussa kuin GPS-formaatissa aina, kun laitteen virta sammutetaan, ja kun virta taas kytketään, järkeillä tallennetusta aikaleimasta, onko viikkonumeron rollover tapahtunut vai ei. Toinen, paljon vähemmän elegantti tapa on vain toivoa, että laitteen käyttöikä ei yllä seuraavaan viikkonumeron kierähtämiseen asti…

Tosin GPS-navigaatioviesteistä on olemassa myös uudempi CNAV-formaatti (vanhempaa kutsutaan LNAV- eli legacy navigation -formaatiksi). CNAV-viesteissä viikkonumero on kasvatettu 13 bittiin, joka riittää jo 157 vuodeksi kerrallaan. CNAV-formaatin tukea markkinoilla olevissa GPS-vastaanottimissa ei selvitetty tätä juttua varten.

GLONASS-, Galileo- ja BeiDou-paikannusjärjestelmillä on kullakin myös omat aikaformaattinsa ja nollakohtansa, mutta emme paneudu niihin tässä jutussa.

Alikehys	Sanan numero		
	1	2	3-10
1	TLM	HOW	Kellonkorjausta, GPS-viikko, satelliitin kunto ym.
2	TLM	HOW	Ephemeris
3	TLM	HOW	Ephemeris
4	TLM	HOW	Almanac, ionosfääritietoa ym.
5	TLM	HOW	Almanac, satelliitin terveys ym.

Taulukko 1. Navigaatioviestin kehyksen rakenne. TLM = telemetry, HOW = hand-over word. TLM alkaa 8 bitin johdannolla 10001011, jota seuraa telemetriaviesti ja 6 bitin Hamming-koodi. HOW koostuu 17 bitin time of week -ajasta, alikehyksen ID-numerosta ja pariteettibiteistä. Time of week tarkoittaa ajanhetkeä seuraavan alikehyksen alussa.

Bitti	Sisältö
1–17	Time of week. Viikon alusta kuluneiden 6 sekunnin jaksojen lukumäärä
18	Alert bit. Jos 1, satelliitin tarkkuus etäisyystarkkuus saattaa olla matalampi kuin alikehyksessä 1 sanotaan.
19	Antispoof bit. Jos 1, satelliitin antispoof-tila on päällä.
20–22	Alikehyksen ID: arvot 1-5.
23–30	Pariteettibittejä.

Taulukko 2. HOW-sanan rakenne.

1. tammikuuta 1601

Muutamissa järjestelmissä ja ohjelmointikielissä ajanlasku alkaa **1. tammikuuta 1601**. Tämä johtunee siitä, että vuosi 1600 on tasavuosisata ja karkausvuosi, mikä varmaankin helpottaa karkausvuosiaritmetiikkaa, sekä myös reippaasti menneisyydessä, eli verrattain vanhojakin päivämääriä voidaan esittää.

Microsoftin Windows NT -käyttöjärjestelmien SYSTEMTIME-ajanlasku alkaa myös 1.1.1601. Tämä pätee NT:n ensimmäisestä vuoden 1993 versiosta aina nykyiseen Windows 11:een asti. Aika tallennetaan sadan nanosekunnin pituisten jaksojen lukumääränä. Samaa formaattia käyttää myös Active Directory.

Cobol-ohjelmointikieli tukee pariakin erilaista aikaformaattia. Yksi niistä on vuoden 1601 alusta kuluneiden päivien lukumäärä. IBM:n *COBOL for AIX 5.1.0* -manuaalin mukaan funktio CEECBLDY muuntaa päivämääräarvon Cobolin kokonaislukuformaattiin, jossa päivä 1 on 1.1.1601. Toinen Cobolin tukema päivämääräformaatti on niin sanottu *Lilian date* eli gregoriaanisen kalenterin alusta kuluneiden päivien määrä, jossa päivä 1 on 15. lokakuuta 1582. Formaatti on nimetty gregoriaanisen kalenterin kehittäneen Aloisius Liliuksen mukaan.

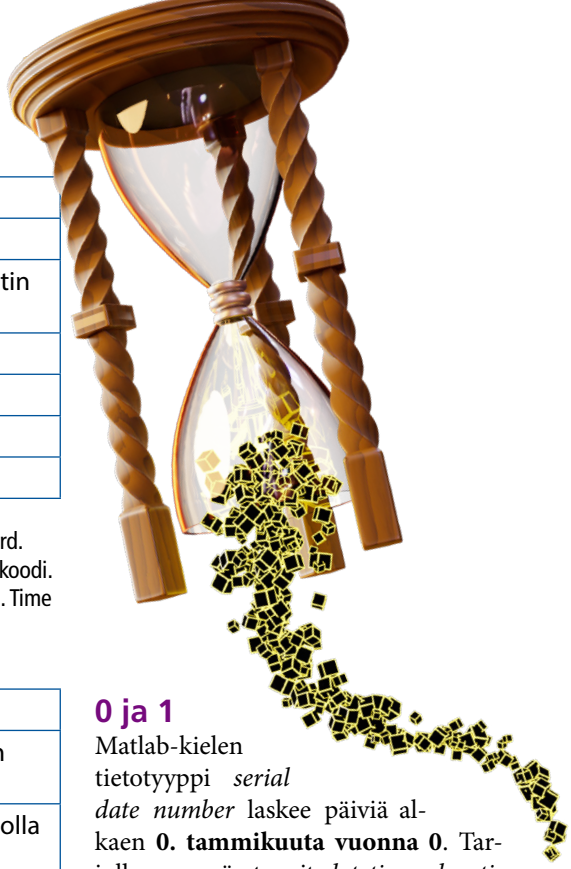
31. joulukuuta 1840 – MUMPS

Massachusetts General Hospital Utility Multi-Programming System onkin Skrollin lukijoille jo tuttu numerosta 2020.4, jossa esiteltiin tätä myös M:ksi kutsuttua ohjelmointikieltä ja tehtiin sillä 3D-grafiikkaa.

Kuten jo tuossa numerossa kerroimme, MUMPS on tietotekniikan aikaskaalalla varsin muinainen, sillä se on julkaistu jo vuonna 1966. Siis kauan ennen esimerkiksi Unix-ajan lanseerausta.

Niinpä MUMPSilla on oma aikaformaattinsa. Päivämäärät tallennetaan nimittäin alkaen **31. joulukuuta 1840** juoksevana kuluneiden päivien lukumääränä ja kellonaika erikseen sekunteina edellisestä keskiyöstä laskien. Päivä 1 on 1.1.1841. Nykyinen päivämäärä on MUMPS-järjestelmässä saatavilla järjestelmämuuttujassa $H (”horolog”).

Päivämäärä on valittu siten, että sillä on voitu varmasti esittää vanhimman mahdollisen hengissä olevan yhdysvaltalaisen ihmisen syntymäaika 1960-luvun lopulla. Valintaan vaikutti myös karkausvuosien kätevä laskettavuus, joten päivä 0 on 31. joulukuuta karkausvuoden 1840 puolella.



0 ja 1

Matlab-kielen tietotyyppi *serial date number* laskee päiviä alkaen **0. tammikuuta vuonna 0**. Tarjolla on myös tyypit *datetime*, *duration* ja *calendarDuration*, jotka toimivat eri tavalla, mutta emme perehdy niihin tässä.

Microsoftin .NET-ohjelmointikielet Visual Basic .NET, C# ja PowerShell laskevat aikaa alkaen **1. tammikuuta vuonna 1**. Samoin tekee Go-kieli. Erona on tosin se, että ajanlaskun resoluutio on .NETissä 100 nanosekuntia, kun taas Gossa peräti yksi nanosekunti.

1899, 1900 ja 1901

Delphi-ohjelmointikielessä on tietotyyppi TDateTime, joka ilmaisee kaksoistarkkuuksisena liukulukuna **30. joulukuuta 1899** jälkeen kuluneet päivät. Siispä 0.00 tarkoittaa 30.12.1899 klo 00.00 ja esimerkiksi 2.50 tarkoittaa 1.1.1900 klo 12.00. Myös negatiiviset arvot ovat tuettuja. [2]

Ilmeisesti Delphin nollakohtavalinnan tarkoituksena on yhteensopivuuden tavoitteleminen Microsoft Excelin kanssa kopioimatta kuitenkaan Excelin virheellistä käsitystä siitä, että vuosi 1900 olisi karkausvuosi.

Excelissä päivämäärät tallentuvat numeroina, jotka vain esitetään käyttäjän toivomassa esitysmuodossa työkirjan soluissa. Päivä 1 on **1. tammikuuta 1900**. Päivä 0 taas on vähän jännästi 0. tammikuuta 1900. Kokeilin myös päivämäärää -1, mutta se ei näyttänyt kelpaavan. Työkaluvihjeessä lukee: ”Negatiiviset tai liian suuret päi-

[2] docwiki.embarcadero.com/Libraries/Alexandria/en/System.TDateTime

vämäärät ja kellonajat näkyvät muodossa #####."

Jos Excel-solun tyyppi on pelkkä päivämäärä, numero on kokonaisluku. Jos taas tyyppinä on päivämäärä ja kellonaika yhdessä, numero onkin desimaaliluku, jonka desimaaliosa on kellonaika esitettynä vajaana vuorokautena prosentuaalisesti, hieman samoin kuin Delphissä. Excel tosin käyttää oikeita desimaalilukuja eikä IEEE 754 -liukulukuja kuten Delphi.

Karkausvuosivirheen vuoksi päivä 60 on Excelin ajanlaskussa 29.2.1900 (jota ei ole olemassa), Delphissä taas 28.2.1900. Maaliskuun 1. päivästä 1901 eteenpäin Delphin ja Excelin sisäiset ajat ovat yhteneväiset, sillä Delphin ajanlasku alkaa yhtä päivää Exceliä aiemmin.

Excel lienee perinyt karkausvuosivirheensä aiemmalta suositulta taulukkolaskentaohjelmalta, Lotus 1-2-3:lta.

IBM:n vanhemmat z/OS-suurtietokoneet laskevat aikaa 64-bittisen *Time of Day clock* -rekisterin (TOD) avulla, joka kuvaa aikaa, joka on kulunut nollakohdasta **1. tammikuuta 1900 klo 00:00:00 UTC**. Aika kasvaa bitin 51 kohdalta joka mikrosekunti. TOD ei ole pelkkä mikrosekuntilukema nollakohdasta alkaen, vaan se sisältää myös esimerkiksi tietokoneen kellon ohjaus- ja korjausdataa kuten *steering rate* ja *episode*, joihin ei perehdytä tässä enempää.

TOD:n 64-bittinen versio loppuu kesken syyskuussa 2042, mutta sen on z/OS:n uudemmissa versioissa korvannut *extended time of day* eli ETOD, joka on kasvatettu 104 bittiin. ETOD-kellossa riittää ajanlaskua vuoteen 36 765 asti.

Ada-kielen ajanlasku alkaa **1. tammikuuta 1901**. Adassa tietotyyppien määrittelyvälit voi ilmaista mielivaltaisesti eikä laskemisen tarvitse alkaa nollasta. Siispä Ada 83:ssa ja 95:ssä kalenterivuosi oli määritelty näin:

```
subtype Year_Number is Integer
range 1901 .. 2099;
```

Ada 2005:stä lähtien katto on nostettu 2399:ään.

1. tammikuuta 2001 – Mac ja iOS

Vaikka nykyajan macOS (aiemmin Mac OS X) ja iOS perustuvat etäisesti BSD-käyttöjärjestelmiin ja ovat siten yhteensopivia Unix-ajan kanssa, niillä on myös oma aikaformaattinsa CFAbsoluteTime.[3] Se on aikamääre, joka on jonkin vertailuajankohdan ja **1. tammikuuta 2001 klo 00.00.00 UTC** välinen erotus sekunteina.

Esitysmuoto on tyyppiä CFTimeInterval, joka on oikeasti kaksoistarkkuuksinen liukuluku (*double-precision floating point*). Mittausresoluutio on alle millisekunnin ja aikaskaalaa riittää ± 10 000 vuotta tuosta vuoden 2001 nollakohdasta.

Vanhassa Mac OS:ssä, joka Mac OS X:n julkaisun yhteydessä nimettiin uudelleen *classic Mac OS:ksi*, aikaa laskettiin 32-bittisenä sekuntilaskurina alkaen **1. tammikuuta 1904 klo 00.00.00**. Tämä laskuri kierähtää ympäri 6. helmikuuta 2040 klo 6.28.15. Käyttöjärjestelmä ei tosin anna käyttäjän asettaa päivämäärää aiemmaksi kuin 1. tammikuuta 1920 tai myöhemmäksi kuin 31. joulukuuta 2019.

MySQL

Relaatiotietokannoissa on lukuisia erilaisia tietotyyppejä, joista aikatyypit ovat aivan keskeisimpiä. Katsotaan esimerkiksi MySQL:ää. Tämä kuvaus pätee (tätä kirjoitettaessa) versioon 5.6.4 ja uudempiin.

YEAR-tyyppi on yhden tavun kokoinen. Siihen mahtuu siis arvot väliltä 0–255. Tämä tyyppi on määritelty välillä 1901–2155, mutta myös arvo 0000 on tuettu. Arvon voi syöttää joko nelinumeroisena tai kaksinumeroisena. Kokonaislukusyöte 00 tulkitaan vuodeksi 0000, mutta merkkijonosyöte '0' tai '00' tarkoittaa vuotta 2000. Kaksinumeroiset syötteet väliltä 0–69 (kokonaislukuna tai merkkijonona) tulkitaan vuosiksi 2000–2069, kun taas syötteet 70–99 tulkitaan vuosiksi 1970–1999.

DATE-tyyppi on kolme tavua (little endian). Se pakataan muotoon vuodet×16×32 + kuukaudet×32 + päivät.

TIME-tyyppi on kolme tavua (big endian) pakattuna seuraavasti:

```
1 bitti etumerkki
(1=ei-negatiivinen, 0=negatiivinen)
1 bittiä varattu
10 bittiä tunti    (0-838)
6 bittiä minuutti (0-59)
6 bittiä sekunti  (0-59)
---------------------------
24 bittiä = 3 tavua
```

[3] developer.apple.com/documentation/corefoundation/time_utilities

DATETIME-tyyppi on viisi tavua (big endian) pakattuna seuraavasti:

```
1 bitti etumerkki
(1=ei-negatiivinen, 0=negatiivinen)
17 bittiä vuosi*13+kuukausi
(vuosi 0-9999, kuukausi 0-12)
5 bittiä päivä     (0-31)
5 bittiä tunti     (0-23)
6 bittiä minuutti (0-59)
6 bittiä sekunti  (0-59)
---------------------------
40 bittiä = 5 tavua
```

TIME- ja DATETIME-tyyppien perässä seuraa vielä sekunnin murto-osia kuvaava tieto, jonka pituus on 0–3 tavua riippuen tietokannan *fractional seconds precision* -asetuksesta.

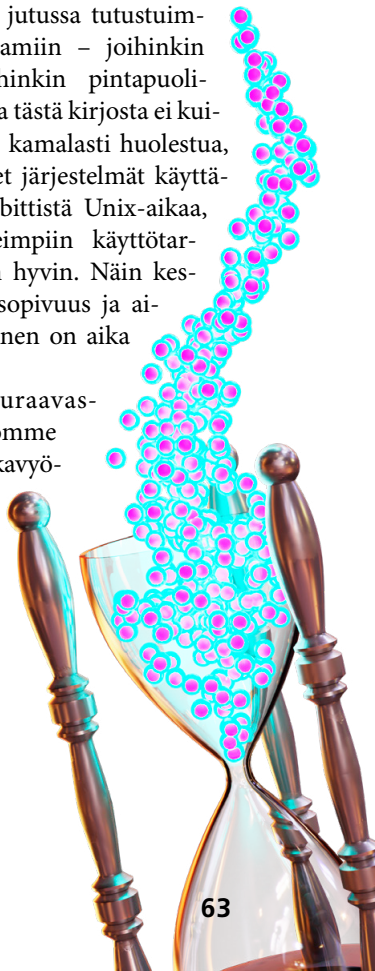
DATE- ja DATETIME-arvot ovat tuettuja välillä 1.1.1000–31.12.9999, TIME-tyyppi taas välillä -838:59:59–838:59:59. TIME-tyypin arvoväli on noin pitkä siksi, että sitä käytetään kellonaikojen lisäksi myös kuluneiden ajanjaksojen ja aikaerotusten ilmaisemiseen.

TIMESTAMP-tyyppi on sama kuin Unix-aika (ks. edellä), nelitavuisena eli 32-bittisenä (big endian). Tämä tyyppi on siis haavoittuva Y2K38-ongelmalle, ellei tyypin bittileveyttä kasvateta tulevissa MySQL:n versioissa.

Kaikki aikanaan

Eri tapoja laskea, tallentaa ja välittää aikatietoja eri tietojärjestelmissä on riittämiin. Tässä jutussa tutustuimme niistä muutamiin – joihinkin tarkemmin, joihinkin pintapuolisemmin. Kaikesta tästä kirjosta ei kuitenkaan kannata kamalasti huolestua, sillä hyvin monet järjestelmät käyttävät nykyään 64-bittistä Unix-aikaa, joka riittää useimpiin käyttötarkoituksiin vallan hyvin. Näin keskinäinen yhteensopivuus ja aikavälien laskeminen on aika selkeää.

Aikasarjan seuraavassa jaksossa kerromme lisää UTC:stä, aikavyöhykkeistä ja karkaussekunneista.



SANA ADASTA

Ohjelmoijan suurennuslasin alla ohjelmointikielet Ada ja SPARK.

Teksti: Henrik Härkönen
Kuvat: Leah Goodreau ja Science Museum Group / Wikimedia Commons

Ohjelmoijana eräs harrastuksistani on erilaisten ohjelmointikielien vertailu ja tutkiskelu. Tutustuminen muistuttaa usein ikkunashoppailua, tyypillisesti en tutustu moneenkaan kieleen kovin syvällisesti. Aina välillä kuitenkin tulee vastaan kieli, jossa on jotain aivan erityistä. Niin erityistä, että on aivan pakko opetella käyttämään sitä. Tällä hetkellä tuo kieli on Ada sekä sen rajoitetumpi versio, alikieli SPARK.

Tyypillistä ohjelmointia

Lähes aina kun mainitsen Adan keskustelussa, kuulijoiden kulmakarvat nousevat ihmetyksestä. Moni ei ole kuullutkaan kyseisestä kielestä, vaikka sen ensimmäinen versio syntyi vain hieman ennen C++ -kieltä. Vähäinen suosio todennäköisesti johtuu siitä, että kieli alunperin kehitettiin Yhdysvaltain puolustusministeriön toimesta korvaamaan lukuisia siellä käytössä olevia sulautettujen ohjelmistojen kieliä. Nykyään tärkeimmät käyttökohteet ovatkin ilmailu- ja avaruusteollisuudessa tai vaikkapa rautateiden valvontajärjestelmissä. Muun muassa Eurofighter Typhoon -hävittäjän järjestelmissä on käytetty Adaa. Kielestä innostuneet harrastajat ovat kuitenkin nykyään jopa hieman harmistuneet kielen saamasta kategorisesta leimasta, vaikka se soveltuu tietysti moneen muuhunkin käyttöön kuin sulautettuihin järjestelmiin.

Edellä mainituista käyttökohteista voi ehkä jo päätellä, että Adaa käytetään tehtäessä ohjelmistoa, jonka on toimittava mahdollisimman luotettavasti. Kielessä on monenlaisia turvaominaisuuksia, joilla pyritään varmistamaan ohjelman moitteeton suoritus. Erityisesti SPARK-versio kielestä mahdollistaa ohjelman todistamisen formaalisti oikeaksi. Ohjelma tai sen kriittisiä osia on mahdollista todistaa korrektiksi siten, että voidaan sanoa, ettei ohjelma kaadu eikä vuoda muistia. Erinomainen ominaisuus varsinkin miljardien eurojen arvoisissa projekteissa.

OTM – Oma Tyyppi Mukaan

Se mikä Adan ominaisuuksista pisti kuitenkin ensimmäisenä silmään oli jotain ohjelmoijalle hyvin arkista: tyyppijärjestelmä. Mielestäni yksi kiehtovimpia asioita ohjelmoinnissa on oikean maailman ilmiöiden ja asioiden mallintaminen tietorakenteilla. Tietorakenteiden pohjalla olevat varsinaiset tiedon muruset, kuten yksittäiset mittaustulokset tai tietokanta-alkiot, kuvataan

erilaisilla tietotyypeillä. Eri ohjelmointikielissä tyyppijärjestelmät vaihtelevat dynaamisesta staattiseen ja rakenteellisesta nimelliseen tyypitykseen. Yhteistä kuitenkin useimmissa kielissä ovat primitiivisten tyyppien ominaisuudet.

Kokonaislukuja voidaan säilöä tyypeillä, joiden nimi on yleensä int tai integer, liukuluvut tallentuvat float- ja double-nimisillä tyypeillä ja niin edelleen. Nimet ja tarkat määritelmät hieman vaihtelevat, mutta pohjimmiltaan kaikki yleensä mukailevat niitä yksiköitä, joissa prosessori tietoa käsittelee. Esimerkiksi Java-kielessä käsiteltäessä alkiota, jonka tyyppi on int, kyseessä on neljän tavun mittainen luku, joka voi saada arvon välillä -2147483648 ja 2147483647.

Kuvitellaan tilanne, missä meillä on mittauslaite, lämpömittari. Se antaa mittausarvoja, joiden pienin arvo voi olla -80 ja suurin arvo +400. Tällöin käsiteltäessä tätä mittausarvoa int-tyyppisessä muuttujassa jää valtava määrä arvoja, joita muuttujaan voi tallentaa, mutta nämä eivät todellisuudessa ole mahdollisia mittauksia. Ne ovat siis virheellisiä arvoja. Tiedon virheellisten arvojen tunnistamiseen on monia keinoja. Se voidaan tarkistaa, kun arvo saadaan ensimmäisen kerran, mutta sallittua arvoa pitää tarkastella myös, jos tiedolla tehdään esimerkiksi laskutoimituksia.

Olio-ohjelmointi tarjoaa toisen tavan hallita tietotyyppiä. Luodaan mittaustulokselle oma tietotyyppi luokkana ja säilötään primitiivinen tyyppi sen sisälle. Luokalle voidaan sitten toteuttaa rajoittuneen arvon tarkistavat operaatiot, joita käytetään aina, kun tietoa muokataan. Valitettavasti käytännössä tällaista käärettä primitiiviselle tyypille tehdään yllättävän harvoin. Hinta uuden luokan tekemiselle voi olla liian suuri. Esimerkiksi Javassa uuden julkisen luokan tekeminen maksaa aina uuden tiedoston ja vähintään muutaman rivin koodia. Monia projekteja läpi kahlanneena käytännön havaintoni on, että kynnys uuden luokan luomiselle tuntuu olevan liian suuri, myös itselleni.

Toinen huono puoli primitiivisten tyyppien yleisessä käytössä on se, että jos funktiossa on monta samantyyppistä parametriä, parametrit saattavat mennä koodatessa sekaisin. Pahim-

millaan ohjelma toimii näennäisen moitteettomasti pitkään, kun arvot sattuvat olemaan sopivia – kunnes ne eivät enää olekaan. Pari kertaa on tällainenkin virhe tullut tosielämässä vastaan, enkä edes itse ollut aina syypää.

Joissakin kielissä, kuten Scala, onkin onneksi mahdollista helposti tehdä tyypeistä aliaksia, joiden avulla tyypille voidaan antaa semanttisesti järkevämpi nimi. Vaikka perimmiltään kyseessä olisikin vain uudelleen nimetty primitiivinen tyyppi, ne ovat kääntäjän mielestä yhteensopimattomia muiden tyyppien kanssa. Tällöin sekaantumisen vaara pienenee.

Edellä mainitut asiat eivät kauheasti häirinneet, kunnes törmäsin Ada-kieleen ja aloin ymmärtää, kuinka siinä tietotyyppejä käsitellään. Sen sijaan, että oma tyyppi nähtäisiin enemmän poikkeuksena, se on pikemminkin sääntö Adassa. Kieli rohkaisee käyttämään omia tyyppejä muun muassa sillä, että niiden määrittely on vaivatonta.

Kuvitteellisen laitteemme mittaustuloksen tyyppi voisi olla seuraavanlainen:

```
type Mittaustulos is range -80 .. 400;
```

Tällöin muuttuja, jonka tyyppi on Mittaustulos, voi sisältää ainoastaan arvoja välillä -80 ja +400. Jos siihen yritetään sijoittaa välin ulkopuolinen arvo, ohjelma nostaa Constraint_Error-poikkeuksen. Myös kääntäjä on tietoinen tästä rajoitteesta ja havaitsee ilmiselvät tilanteet, joissa arvon on mahdollista olla virheellinen, eikä virheen havainnointia tarvitse odottaa ajonaikaisesti.

Jollakin välillä (*range*) olevan arvon määritteleminen on erikoistunut tyyppirajoite. Rajoitteen voi kirjoittaa yleisempään muotoon with-ilmaisulla. Esimerkiksi tyyppi, joka sallisi vain parillisia kokonaislukuja, olisi seuraavanlainen:

```
type    Even is new Integer
with    Dynamic_Predicate => Even mod 2
= 0;
```

Väli voi myös määritellä taulukon alkioiden indeksin. Adassa taulukoiden indeksin ei tarvitse olla nollasta eteenpäin, vaan se voi alkaa numerosta 1 tai vaikka -300. Myös enumeraatiot voivat olla taulukon indeksinä. Tämä helpottaa mallintamista, koska taulukoita ei tarvitse sovittaa alkamaan nollasta, kuten monissa C:n kaltaisissa kielissä. Nolla-alkuisten taulukoiden käsittelyssä esiintyykin aivan oma virhekategoriansa, jota kutsutaan lempinimellä ”off by one -error”.

Edellä olevan kuvitteellisen laitteen mittaustuloksista voidaan helposti kirjata histogrammi määrittelemällä mittauksille taulukko, jonka ensimmäinen indeksi on -80 ja viimeinen 400.

```
type Histogrammi_Type is array
(Mittaustulos) of Natural;
Histogrammi : Histogrammi_Type;
```

Tällöin esimerkiksi -80 mittaustuloksen saatuamme voimme kasvattaa kyseisen tuloksen histogrammiarvoa:

```
Histogrammi(-80) := Histogrammi(-80)
+ 1;
```

Mutta miten tiedetään taulukon rajat, jos väli voi olla mikä vain? Monilla asioilla Adassa on attribuutti, johon viitataan heittomerkillä. Histogrammi’First kertoo ensimmäisen alkion indeksin ja Histogrammi’Last viimeisen. Koko taulukko voidaan käydä läpi for-silmukassa.

```
for I in Histogrammi’First ..
Historgrammi’Last loop
Put_Line (Histogrammi (I)’Image);
end loop;
```

Esimerkki tulostaa kaikki histogrammin arvot. Histogrammi’First .. Histogrammi’Last voidaan myös lyhentää muotoon Histogrammi’Range. Attribuutteja käyttämällä varmistamme, ettei ohjelma nosta poikkeusta taulukon ylittämisen johdosta. Image-attribuutti taas antaa arvon esityksen tekstimuodossa.

Ada myös varmistaa, että ohjelmoija varmasti tarkoittaa sitä, mitä on kirjoittamassa. Kielessä ei tarkoituksenmukaisesti ole epäsuoria tyyppimuunnoksia. Esimerkiksi liukulukua ei voi jakaa kokonaisluvulla. Tyyppimuunnos on tehtävä selväksi koodissa. Samasta syystä kielessä ei myöskään ole tyyppipäättelyä, vaan jokainen tyyppi pitää erikseen määritellä muuttujille.

Osoittelematta paras

Esimerkiksi C-kielessä monen virheen syy on muistiosoittimien virheellinen ohjelmointi. Osoitinaritmetiikka tarkoittaa, että käsiteltävien muistialueiden osoitteita voidaan laskea yhteen tai vähentää, jotta alueen alkioita voi lukea tai kirjoittaa. Sallitun muistialueen yli kirjoittaminen voi aiheuttaa vaikeasti havaittavia ongelmatilanteita. Adassa huolimattoman osoitinaritmetiikan aiheuttama muistin yli kirjoittaminen on vaikeaa, koska se ei salli sitä. Osoittimet ovat access type -tyyppimäärittelyn takana, ja niiden väärinkäyttö vaatii ohjelmoijalta aivan erityistä ponnistelua.

Osoitintyyppejä tarvitsee kuitenkin yllättävän harvoin, koska Adassa funktio ja proseduuri voivat palauttaa taulukon suoraan, ilman osoittimia. Taulukko voi tietenkin olla myös parametrina. Käytännön toteutus jää kääntäjän murheeksi.

Myös yksittäisten muuttujien tapauksessa osoitintyyppiä tarvitaan harvoin, koska parametrit voivat olla erikseen joko proseduuriin sisään tulevia arvoja, ulos tulevia arvoja tai molempia yhtä aikaa. Tyyppi määritellään in-, out- tai in out -merkinnällä parametrin nimen yhteydessä.

Tietenkin osoittimet ovat hyvin hyödyllinen ominaisuus, jota käytetään Ada-ohjelmissakin paljon, mutta yllättävänä havaintona on, että hyvin paljon voi saavuttaa myös ilman niiden jatkuvaa käyttöä.

- Ada on ollut ISO-standardi sen ensimmäisestä versiosta, Ada 83:sta, lähtien, ja monista muista ISO-standardeista poiketen se on vapaasti saatavilla. Kieltä on päivitetty vuosina 1995, 2005 sekä 2012. Uusin versio on parhaillaan tekeillä, ja sen nimeksi tulee Ada 2022. Osa uusista mukaan tulevista ominaisuuksista on jo käytettävissä GNAT-kääntäjässä erityisillä kääntäjävivuilla.
- AdaCore-sivusto, jossa voi opetella: learn.adacore.com
- Eräs aktiivinen Ada-yhteisö on Gitter.im-keskusteluohjelmassa
- huoneessa ”ada-lang/Lobby”, joka on sillattu myös Matrix-pikaviestimeen: gitter.im/ada-lang
- Adalle on myös oma sub-reddit: reddit.com/r/ada
- Alire-paketinhallinta: alire.ada.dev

Alire – työkalupakin päivitys

Aluksi omaa Ada-opiskeluintoani hillitsi hieman se, että Adan työkaluket-

```
with Ada.Text_IO; use Ada.Text_IO;

procedure Skrolli is

        type Mittaustulos is range -80 .. 400;
        type Histogrammi_Type is array (Mittaustulos) of Natural;

        subtype Even is Integer
                with Dynamic_Predicate => Even mod 2 = 0;

        -- alustetaan taulu nollalla
        Histogrammi : Histogrammi_Type := (others => 0);

        -- aliohjelma aliohjelman sisällä
        procedure Nollaa (Taulu : in out Histogrammi_Type) is
        begin
                for I in Taulu'Range loop
                        Taulu (I) := 0;
                end loop;
        end Nollaa;

begin
        Histogrammi(-80) := Histogrammi(-80) + 1;
        Parillinen := 7; -- virheellinen arvo, joka jää kiinni jo käännöksessä!

        Nollaa(Histogrammi);

        -- tulostetaan histogrammin kaikki arvot
        for I in Histogrammi'Range loop
                Put (Histogrammi(I)'Image);
        end loop;

end Skrolli;
```

justa puuttui moderni paketinhallinta. Kaikki ohjelmakirjastot piti hankkia ja kääntää itse ja lisätä projektiin. Asiaan on kuitenkin hiljattain tullut erittäin toivottu muutos. Ohjelmakirjastoja ja niiden riippuvuuksia voi hallita nykyään ALIRE (*Ada LIbrary REpository*) -projektin tarjoamalla alr-komentoriviohjelmalla. Oman projektin tarvitseman kirjaston voi asentaa yhdellä komennolla, jolloin Alire noutaa lähdekoodin esimerkiksi GitHubista ja asentaa projektin alihakemistoon niin, että se kääntyy samalla kuin oma projektikin.

Alire lainaa termistöä Rust-kielen vastaavalta työkalulta, ja ohjelmakirjastot ovat crate-nimisiä yksiköitä. Näiden laatikoiden määrä on toistaiseksi vaatimaton, mutta AdaCore-niminen yritys, joka on GNAT-kääntäjän ja työkaluketjun yksi suurimpia kaupallisia hyödyntäjiä ja samalla avoimeen lähdekoodiin lahjoittavia tahoja, järjesti vuonna 2021 ensimmäistä kertaa *Crate of The Year* -kilpailun, jossa palkintona oli kolmessa eri kategoriassa 2 000 dollarin palkinnot parhaasta ”laatikosta”. Laatikoiden määrä hieman kasvoikin kilpailun myötä.

Valitettavasti oma ensimmäinen laatikkoni on niin vaatimaton, etten tohtinut osallistua kisaan, mutta ehkäpä ensi kerralla! Varmaksi voi kuitenkin sanoa, että matkani Adan parissa tulee jatkumaan.

Matemaatikko **Ada Lovelace** (**Ada King**, Lovelacen kreivitär, 1815–1852) työskenteli **Charles Babbagen** Analytical Engine -koneen parissa, ja häntä pidetään usein maailman ensimmäisenä ohjelmoijana. Lovelace kirjoitti ensimmäiset tietokoneohjelmat paperille täydentäessään analyyttisesta koneesta tehtyjä muistiinpanoja. Ada-ohjelmointikieli on nimetty hänen mukaansa. Muotokuvataiteilija mahdollisesti **Alfred Edward Chalon**.

RastPort KA59

Mekaaninen näppäimistö Amiga 1200:lle

Jos saat jostain käsiisi Commodoren tai myöhempien Amiga-yhtiöiden peruja olevan Amigan piirisarjan,voit nyt rakentaa itsellesi ihan kokonaisen uuden ja alkuperäistä tukevamman A1200:n. Puolalaista suunnittelua oleva näppäimistö täydentää paletin.

Teksti ja kuvat: Mikko Heinonen

Rakennettuani oman ReAmiga 1200:n (Skrolli 2020.3) ja kääräistyäni sen A1200.netin uusiotuotantokoteloon minua jäi harmittamaan enää yksi asia – nimittäin se, ettei minulla ollut koneelle omaa näppäimistöä. Rosvosin piirisarjan tornikoteloon asennetusta koneesta, jolle oli käynyt ulkovarastossa kalpaten, eikä sen alkuperäisiä komponentteja ollut enää juurikaan jäljellä.

Amiga 1200:n alkuperäiset näppäimistöt valmisti Mitsumi, ja toisin kuin vielä A600:n tapauksessa, käytössä oli vain yhtä mallia. Se perustui tuolloin tyypilliseen toteutukseen eli muoviseen kalvoon, jonka pinnalla kulkevia johtimia yhdisteltiin muovisten tappien päässä olevilla kontakteilla. Näppis ei ollut tuohon maailmanaikaan mitenkään erityisen huono, jos ei hyväkään, mutta vanhemmiten se kärsii taudista, jossa kontaktit hapettuvat ja vaativat aina vain enemmän painallusta, jotta näppäinkoodi menee läpi ohjaimelle asti. Asiaa saa autettua puhdistamalla kalvoa mekaanisesti (ei alkoholilla, sillä se voi liuottaa muovia!), mutta lopulta kontaktit kuluvat ja näppäimistö ei enää toimi luotettavasti.

Ensiavuksi ongelmaan ovat tulleet useat harrastajat, jotka ovat suunnitelleet ja valmistuttaneet uusia kalvoja näppäimistöön. Tällaisen uusiotuotteen avulla on myös mahdollista muuttaa paljon yleisemmän Amiga 500 -mallin näppäimistö toimimaan A1200:ssa, sillä itse näppäimistön fyysinen koko ja muoto ovat aivan samanlaiset. Vain pohjan metallilevyyn on kohdistettava hieman työstötoimenpiteitä.

Tämä auttaa ongelmaan, mutta ei ratkaise sitä kokonaan. Ensinnäkin myös ehjät Amiga 500:n näppäimistöt alkavat jo olla kiven alla, ja toisekseen nuorimmillaankin 90-luvulta peräisin oleva, useimmiten runsaasti käytetty ja komeasti kellastunut näppäimistö ei näytä kovin edustavalta uuden emolevyn ja kotelon parina. Uusia näppäinhattuja on ollut tekeillä joukkorahoitusprojektissa, josta on kuitenkin tullut hyvä esimerkki huonosta esimerkistä. Rahoituksen päättymisestä kun on kulunut jo puoli vuosikymmentä, eikä kaikilla osallistujilla (minä mukaan lukien) ole edelleenkään niitä nappuloita. Virallisesti hommaa ei toistaiseksi ole peruttu.

Tehdään itte parempi

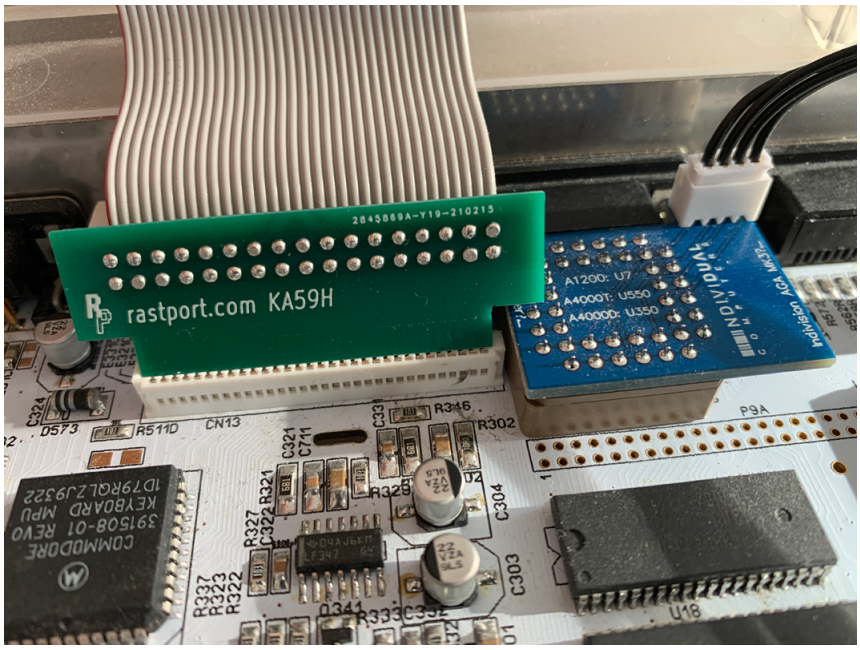
Ei siis ole ihme, että harrastajakunnassa on virinnyt useampiakin hankkeita kokonaisen uuden näppäimistön valmistamiseksi. Se ei ole laitteena hirvittävän monimutkainen, sillä varsinainen ohjainpiiri sijaitsee A1200:n emolevyllä. Piirilevyjen valmistami-

nen taas on nykyisin sen verran edullista, että niitä voidaan tehdä pieninäkin sarjona. Mekaanisten näppäimistöjen suosion kasvun myötä saatavilla on paljon erilaisia valmiita kytkimiä, joilla laitteen voi toteuttaa helpommin kuin muovikalvolla, sillä yläosaa ja kytkinten varsia ei tarvitse valaa erikseen. Bonuksena saadaan parempi näppäintuntuma ja luotettavuus.

Ihan täysin valmista ratkaisua ei kuitenkaan kengännauhabudjetilla koota, sillä jäljelle jäävät vielä Amigan näppäimistön erityispiirteet. Välilyönti on huomattavasti leveämpi kuin normaalissa PC-yhteensopivien näppäimistössä, ja Del- ja Help-näppäimet taas vievät leveyssuunnassa puolentoista normaalin näppäimen tilan. Käytännössä täydellisen kloonin valmistaminen vaatisi uusien näppäinhattujen ruiskuvalamisen, mikä on juuri se kipupiste, johon Kickstarter-näppäinhatut ovat jumittuneet. Prosessi vaatii kalliit muotit, joiden kuolettamiseksi hattuja pitää tehdä pitkä sarja, ja valmistuskapasiteetin saaminen on osoittautunut haasteelliseksi. Näin ollen järkevämpi lähestymistapa on käyttää vakiokokoisia näppäimiä.

Puolalainen **Grzegorz Kraszewski** lähti liikkeelle juuri tästä. Hänen valmistamansa RastPort KA59 on Amiga 1200:lle tarkoitettu mekaaninen näppäimistö, joka perustuu tunnettuihin Cherry-kytkimiin. Se korvaa kokonaan alkuperäisen näppäimistön, joten mitään elimenluovuttajaa ei enää tarvita. Näppäimistöjen ennakkotilaus käynnistyi vuoden 2021 syyskuussa, ja juuri vuoden vaihduttua sain käsiin oman laitteeni, jonka sarjanumero on 12. Halusin sen mustilla näppäinhatuilla ja hiljaisilla Cherry Brown -kytkimillä, sillä en erityisemmin pidä mekaanisten näppäimistöjen kovaäänisestä naksutuksesta. Näppäimistön asettelu on ihan oikea FI/SWE, eli mukana ovat meikäläiset ääkköset. Tätä helpottaa osaltaan se, että A1200:n näppäimistöä tehtiin Eurooppaan vain yhtä mallia, jolloin perustason UK-asettelussa siinä on tyhjiä näppäimiä. Hattuihin on tehty tekstit siististi painattamalla eikä tarrasarjalla, joita on nähty kotitekoisemmissa ratkaisuissa.

Kuten kuvista käy ilmi, KA59 sisältää normaaliin Amigan näppäimistöön verrattuna kolme ylimääräistä näppäintä. Altien molemmin puolin on ”RastPort”-näppäimet, Delin ja Helpin välistä taas löytyy ”Extra”. RP-näp-



päinten toiminnon voi valita muutaman vaihtoehdon välillä muuttamalla piirilevyllä olevia siltauksia – joskaan kirjoitushetkellä tästä ei ole vielä kovin selkeää ohjetta, joten en tarkalleen tiedä miten. Extra sen sijaan on yhdistetty vain näppäimistön piirilevyn reunassa olevaan riviliittimeen. Mieleeni tulee, että sitä voisi käyttää vaikkapa ohjaamaan näyttösignaalin vaihtoa tai USB-levyasemaemulaattoria.

Pientä ekstraa on sekin, että Caps Lockin merkkivalon väriä voi säätää piirilevyn reunassa olevista RGB-potentiometreistä. Kaikki tuskin koskaan tätä käyttävät, mutta se on mukava kädenojennus sen suuntaan, että Amigoihin on valmistettu viime vuosina myös monia erivärisiä kuoria ja jopa väriään vaihtavia virtavaloja.

Pieni kauneusvirhe

KA59 vaikuttaa kokonaisuutena, ja etenkin pieni valmistusmäärä huomioiden, laadukkaalta tuotteelta. Piirilevy on koottu huolellisesti ja kasattu ruuveilla, joten näppäimistön voi avata vaikkapa kytkimen vaihtoa varten. Rakenne on napakka, vaikkei se sisällä metallista pohjalevyä. KA59 istuu kauniisti paikoilleen ainakin A1200.netin koteloon, eikä ole syytä epäillä, etteikö tilanne olisi sama Commodorenkin kopassa.

Pikkuinen ongelma tuli kuitenkin vastaan oman ReAmigani Indivision Mk3 -näyttöadapterin kanssa. Siihen kuuluu lisäpiirilevy, joka asetetaan toisen CIA-piirin päälle ja jonka kautta signaalimuunninta voidaan ohjata itse koneesta riippumatta. CIA U7 on emolevyllä aivan näppäimistöliittimen vieressä, ja RastPortin käyttämä, sinänsä laadukas ja tukeva liitäntäpiirilevy taas on valitettavasti niin leveä, että se ei mahdu paikoilleen, jos Indin sovitin on asennettuna.

Indivision kyllä toimii ilmankin tätä piirilevyä, mutta sen käyttö hieman vaikeutuu. Ongelma on myös mahdollista korjata asentamalla CIA:n päälle matala PLCC-kanta ja juottamalla johtimet suoraan siihen, sillä kytkentäkaavio löytyy Individual Computersin wikistä. Tämä vaatii kuitenkin sen verran harrastuneisuutta, että otan asian puheeksi valmistajan kanssa – silmämääräisesti tutkien levyä saisi pienennettyä niin, ettei se peitä CIA:ta aivan näin paljon.

Tämä onkin sitten ainoa negatiivinen asia, jonka KA59:stä tuotteena keksin. Sen tuntuma on selvästi parempi kuin Mitsumissa ja kokoonpanon laatu on korkeaa. Noin 200 euron hinta on sekin kohtuullinen kun ajatellaan, että uudenveroinen varaosa maksaa jo harrastajien välisessä kaupassa lähes saman, eikä niitä useinkaan ole tarjolla juuri halutulla näppäimistöasettelulla.

Suomessa RastPortin tuotteita edustaa AmiGarage, www.amigarage.fi. KA59:t valmistetaan tilauksesta, joten on syytä varautua pieneen odotteluun. Kirjoitushetkellä hinta on 197,90 €, mutta epävakaa maailmantilanne voi luonnollisesti muuttaa sitä. Jutussa esiteltävä näppäimistö on ostettu normaalisti täyteen hintaan.

Kuten yleensä käy, Grzegorz ei ole ollut hankkeensa kanssa yksin liikkeellä. Osoitteesta amigakb.wordpress.com löytyy toinen projekti mekaanisen näppäimistön sovittamisesta Amigalle. Se on hieman yleismallisempi, mutta ei tarjoa kuin yhden asettelun ilman skandinaavisia näppäimiä. Näitä näppäimistöjä on myyty noin 160 euron hintaan, mutta ilmeisesti kysyntä on ollut niin kovaa, että tilauksia ei tällä hetkellä oteta vastaan. Myös ainakin Pohjois-Amerikan puolella vaikuttava **kipper2k** on tehnyt oman näppäimistönsä.

Haastattelussa Grzegorz Kraszewski

Ostaessani näppäimistöä AmiGaragesta tuli kauppias **Tapio Koivuniemen** kanssa puheeksi, että ajattelin kirjoittaa laitteesta jutun lehteen. Hän lupautui ystävällisesti kysymään, haluaako suunnittelija kertoa hieman hankkeen taustoista, ja Puolasta tuli ripeästi myönteinen vastaus. Tässä siis suunnittelijan kertomusta omasta päätymisestään elektroniikan pienvalmistajaksi.

Millainen taustasi on, miten päädyit elektroniikka-alalle?
Rakensin ensimmäisen virtapiirini 10-vuotiaana; se oli joulukuusen tähti, jossa oli vilkkuvia LED-valoja. Sen jälkeen aloin tehdä monimutkaisempia laitteita ja parannella nykyisiä. Lisäsin esimerkiksi stereodekooderin kotonamme olleeseen radiovastaanottimeen. Hakeuduin harrastustani vastaavaan koulutukseen, joten opiskelin ensin elektroniikkaa toisella asteella ja suoritin sitten viisivuotiset elektroniikan diplomi-insinööriopinnot Białystokin teknillisessä korkeakoulussa.

Milloin päätit aloittaa lisälaitteiden rakentamisen Amigalle?
Vuonna 1998. Silloin oli paljon keskustelua PCI-korttien liittämisestä Amigaan. Aloitin silloin AmiPCI-nimisen projektin, joka oli hieman naiivikin yritys liittää yksittäinen PCI-kortti Amiga 1200:n laajennusväylään. Saatuani aikaan toimimattoman prototyypin luovutin hankkeen kanssa muutaman kuukauden kuluttua. Vuonna 2000 minuun otti yhteyttä Matay-niminen yritys, joka halusi tehdä Amiga-laitteistoa ja etsi suunnittelijaa. Teimme sopimuksen Amiga 3000/4000 -malleihin tarkoitetusta PCI-sillasta, jonka nimeksi tuli Prometheus.Kun Prometheus saatiin julkaistua, kiinnostuin MorphOSista (ja muutenkin uuden sukupolven Amigoista) ja lopetin elektroniikan suunnittelun yli 10 vuodeksi. Koska minulla on vankka tausta Amigan ohjelmoinnista, liityin MorphOS-tiimiin ja avustin osaltani järjestelmän kehitystä.Palasin Classic-Amigoiden pariin vuonna 2015, kun ymmärsin, ettei MorphOS enää sovellu ensisijaiseksi käyttöjärjestelmäkseni. Siirryin Linuxiin ja… ostin Amiga 1200:n. Lisäsin siihen ensin kiintolevyn SD2IDE-muuntimella ja havaitsin, että kiintolevyn valo palaa jatkuvasti. Suunnittelin pienen piirilevyn, joka ratkaisi ongelman ja toimi myös kiinnikkeenä muuntimelle, jolloin lattakaapelia ei enää tarvittu. Näin syntyi tuote nimeltä KA47. Tarvitsin vain yhden, mutta piirilevyjä piti tehdä kymmenen, joten tarjosin niitä myyntiin paikallisella foorumilla. Kaikki 9 myytiin viikossa, joten teetin toisen erän, ja näin Amiga-raudan valmistus sitten alkoi.

Miksi valitsit juuri Amigan?
Multimediaominaisuuksien vuoksi. Omistin Atari 65XE:n ja harkitsin Atari ST:n hankintaa, mutta kaksi tuttavaa korkeakoululta esittelivät minulle omia koneitaan, 386-PC:tä ja Amiga 1200:aa. Päätin, että Amiga on grafiikaltaan ja ääniltään parempi ja myös edullisempi hankkia. Amiga 1200:aan ei ollut varaa, joten ostin käytetyn A500:n ja laajensin sitä myöhemmin kiintolevyllä ja fast-muistilla. Aloin tutustua Amigan käyttöjärjestelmään ensin käyttäjänä ja sitten ohjelmoijana, ja se oli mahtava kokemus.

Kauanko kesti kehittää Amiga-näppäimistö ideasta valmiiksi tuotteeksi?
Kolme vuotta. Aloitin helmikuussa 2019 ja ensimmäinen prototyyppi valmistui saman vuoden toukokuussa. Sitä seurasi kuusi prototyyppiversiota, joista viimeisin (rev. G) oli ensimmäinen myyntikelpoinen. Nyt olen versiossa J (10.) ja se vaikuttaisi olevan hyvä.

Mitkä olivat suurimmat haasteet (jos nyt ei huomioida koko maailmaa koskevia)?
Logistiikka tekee kaikesta todella hidasta. Totta kai oli myös suunnitteluun liittyviä ongelmia, mutta ne olivat ratkaistavissa: esimerkiksi näppäinhattujen painatus ja näppäimistön oikea jäykkyys. Jälkimmäinen on ratkaistu hiilikuituprofiilin avulla, joten tämä on todellinen high-tech-tuote ;-).

Onko menekki ollut odotettua vai oletko yllättynyt suuntaan tai toiseen?
Arvioin, että pääsisin omilleni eli saisin kulut katettua myymällä 25 näppäimistöä. On ollut mukava nähdä, että kysyntää on ollut reilusti enemmän.

Skrolli kiittää Grzegorzia haastattelusta ja Tapiota sen mahdollistamisesta.

Kännyköiden kiertoviestejä metsästämässä

Kaikki on katoavaista ja kaikki digitaalinen erityisen katoavaista. Muutama vuosi sitten yritin selittää alakouluikäiselle lapselleni tekstiviestianimaatioiden konseptia.

Teksti: Vesa Linja-aho

Koska nykypuhelimilla voi lähetellä ilmaiseksi kaikille lähes mitä tahansa gif-animaatioista pitkiin videoihin, eikä mitään ascii-grafiikkana piirrettyjen animaatioiden tapaistakaan liity arkielämään, selittämistaidot olivat koetuksella. Internetin syövereistäkään ei löytynyt esimerkkiä selityksen tueksi. Merkkigrafiikalla piirretyt kuvat olivat onneksi tuttu käsite, joten sain jotenkin kuvailtua konseptin, jossa pienelle näytölle piirrettiin merkkigrafiikalla yksinkertainen kuva, joka ikään kuin liikkui kun tekstiä vieritti alaspäin. Itse en muista tällaista animaatiota ikinä taiteilleeni, koska työläys oli kertaluokkaa suurempi kuin vihkon nurkkaan piirrettävän animaation rustaamisessa kynällä. Läheteltyä niitä kyllä tuli.

Poro on parasta

1980-luvulla syntyneet ja sitä vanhemmat muistanevat erilaiset huumoritekstiviestit, jotka kiersivät kaverilta kaverille. Yleensä sisältönä oli kaksimielinen vitsi tai loru. Alapääjutut olivat keskeinen teema myös näissä merkkigrafiikka-animaatioissa, vaikka mukana oli toki söpöjä nalleja tai synttärikakkujakin.

Omien aivojen perukoilta muistan esimerkiksi animaation, jossa mies yhtyy poroon takaapäin. Viimeisellä ruudulla luki ”poro on parasta”. Kyseiseen tekstiviestianimaatioon viitattiin myös 2000-vuosikymmenen loppupuolella julkaistussa Nyt-liitteen artikkelissa, jossa käsiteltiin naisten kokemuksia ahdistelevista autokoulunopettajista. Eräs ajo-oppilas oli saanut opettajaltaan vapaa-aikanaan tekstiviestejä, joista yksi oli tämä mainittu poroanimaatio. Oppilas oli vastannut vaivautuneena ”kiva koira”, johon opettaja oli vastannut ”Jumalauta nainen, se on poro!”

Toisessa muistamassani animaatiossa tikku-ukko (tai oikeastaan -akka) tulee kotiin ja yllättää kumppaninsa itsetutkiskelupuuhista, ja käydään seuraava keskustelu:

– Moi, tulin…
– Hyi vittu, oot hanskalla!
– Hyvästi, pervo.

Tätä viestiä muistellessa herää myös kulttuurihistoriallinen pohdinta: ihanko oikeasti joku jättäisi kumppaninsa yllätettyään hänet masturboimasta, ja vielä päälle kutsuisi pervoksi? Samankaltaisia pohdintoja aiheutti, kun luin pari vuotta sitten kohua julkaisuaikaansa herättäneen **Harri Sirolan** nuortenkirjan Abiturientti (1980), jossa yllätykseksi pieneskely oli kuvattu nuorten seurustelusuhteiden kammottavimmaksi tabuksi. Samaa sarjaa on myös 1990-luvun alussa lukemani koululaisvitsi, jossa lapsi rukoilee iltarukouksessa isän pöytälaatikossa olevien lehtien naisille rahaa vaatteisiin, minkä jälkeen vaimo ottaa eron. Vitsit ovat vitsejä, mutta kuvaavat myös jollain tasolla aikansa arvomaailmaa.

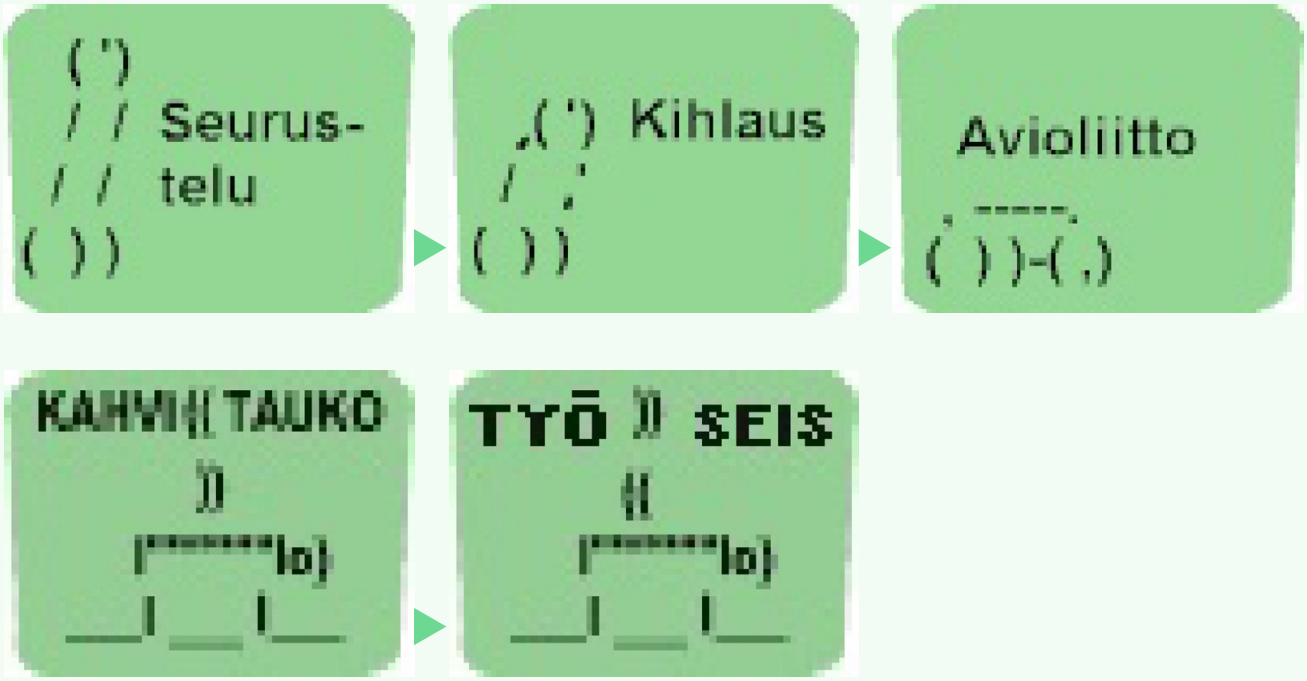
Kolmannessa mieleen jääneessä kiertoviestissä oli ascii-grafiikalla piirretty nainen, ja tarina meni suurin piirtein näin: ensimmäisessä ruudussa kurvikkaan naisen kuvan vieressä oli ikä 20 v, toisessa ruudussa oli hieman pulskempi nainen (30 v) ja kolmannessa oli räjähdyksen kuva ja teksti BOOM. Viimeisellä ruudulla luki vain ”Vaihda ajoissa.” Nykyään moinen huumori ei ole sosiaalisesti hyväksyttävää – tai vähintään pitää miettiä, kenen seurassa sitä viljelee. Toisaalta kiertotekstiviestit eroavat sosiaalisesta mediasta siten, että tekstiviestien kanssa oli jo teknis-taloudellisista syistä pakko soveltaa harkintaa sen suhteen, kenelle niitä lähetti. Instagram-postaus näkyy niin kaverille, pomolle kuin äidillekin. Sovinistisen tekstiviestin saattoi lähettää vain valitulle äijäporukalle, hintaan 1 markka per äijä.

Sopimattomasta huumorista on luonteva jatkaa kiertotekstiviestiin, joka päättyi sanoihin ”Terveisiä Jololta!” Tämän muistan elävästi, koska en jonkun aivojen oikun ansiosta saanut ensin selvää tätä edeltävästä animaatiosta, ja kummeksuaessani asiaa naispuoliselle kaverilleni hän valisti heti, että siinä on penis, jonka päälle laskeutuu takapuoli. Siis epäasiallinen viittaus aikansa mediatapaukseen eli **Risto Vahasen** ja **Seppo Fräntin** panttivankidraamaksi muuttuneeseen lomareissuun.

Mutta onko niitä tallessa?

Suomalaisten historian säilyttämisorganisaatioiden arkistoista kiertotekstiviestejä ei vaikuta löytyvän. Mediamuseo Rupriikin tutkija **Outi Penninkangas** kertoo, että ainakaan heidän museollaan ei ole niistä tallella materiaalia. Vesiperän tuottaa myös laajasta Nokia-kännykkäkokoelmastaan (Skrolli 2020.1) tuttu **Rasmus Hakala**, joka ottaa tietosuojan vakavasti ja on tyhjentänyt kokoelmansa puhelimet vanhoista viesteistä ja yhteystiedoista. Hakalan kautta varmistuu myös animaatiokulttuurin vahva sidos aikalaisuuteen: Hakala oli Nokia 3310:n julkaisuvuonna kuusivuotias, ja hänelle koko tekstiviestianimaation käsite on vieras. ”En muista lapsuudessani tai nuoruudessani törmänneeni noihin animaatioihin lainkaan”, Hakala kertoo.

Puusta pidemmälle eivät johda myöskään yhteydenotot yliopistotutkijoihin. Väitöskirjassaan kännykkäkulttuuria tutkinut Helsingin yliopiston humanistisen tiedekunnan kulttuuri-



Esimerkkejä SMS-animaatioista. Koska Nokian puhelimet vierittivät viestiä koko ruudun kerrallaan (ei siis riveittäin), merkkigrafiikalla pystyi piirtämään ascii-animaation. Esimerkit sivulta niko_85.tripod.com/tekstiviestit/animaatiot.htm.

osaston yliopistonlehtori **Saila Poutiainen** ei muista aiheesta olevan tutkimusta, mutta hän listaa muutaman artikkelin, jotka kannattaa käydä läpi. Mielenkiintoisia artikkeleita kaikki, mutta yksikään ei tuonut lisävaloa juuri tekstiviestianimaatioasiaan.

Turun yliopiston digitaalisen kulttuurin professori **Jaakko Suominen** vinkkaa tutustumaan **Sonja Kankaan** ja **Kaisa Cooganin** tutkimusraporttiin Nuoret ja kommunikaatioakrobatia. 16-18-vuotiaiden kännykkä- ja internetkulttuurit vuodelta 2001. Sielläkin ilmiötä vain raapaistaan – ja sanamuoto on sellainen, että jos ei ennestään tiedä mistä on kysymys, koko konsepti jää epäselväksi:

Vielä nuoria kuitenkin viihdyttävät ”virusmaiset tekstiviestit”, joita piireissä leviää. Näitä ovat mm. ”puhelimessasi on hikka”, jossa näyttö välkkyy ja ilmoittaa puhelimen saaneen hikan. Tällaiset viestit kiinnostavat, koska nuoret eivät keksi kuinka ne ovat tehty. Kuvaviestit taas eivät sykähdytä: kuvat ovat onnettomia ja niiden lähettäminen maksaa moninkertaisesti tavalliseen tekstiviestiin verrattuna. Kuvasta ei nähdä saatavan yhtään suurempaa iloa kuin tavallisesta tekstiviestistä.

Viraalia kutsuttiin siis kaksi vuosikymmentä sitten virusmaiseksi. Hikkaviestin olemassaolon muistan hämärästi, tarkkaa sisältöä en. Joissain puhelinmalleissa toimi vilkkuva fontti, olisiko tämä tehty sillä? Vastaavanhenkinen pintaraapaisu löytyy myös **Eija-Liisa Kasesniemen** ja **Pirjo Rautiaisen** tutkimuksesta Kännyssä piilevät sanomat (2001), jonka aineistona olivat nuorten paperilomakkeille keräämät tekstiviestit. Osallistuin kyseiseen tutkimukseen lukiolaisena ja muistan, kuinka paperilomakkeelle viestien kirjoittaminen oli puisevaa hommaa. Muistaakseni kieltäydyin suosiolla täyttämästä lomakkeelle näitä animaatioviestejä, koska niistä ei kerta kaikkiaan edes saanut selvää, koska lomakkeella oli erilainen rivitys kuin puhelimen näytöllä. Tutkijat valittelevat samaa:

Lisäksi osa viesteistä on erittäin vaikeita kopioida paperille. Tällaisia hankalia viestejä ovat esimerkiksi animaation kaltaiset kuvaviestit sekä viestit, joissa yksi merkki toistuu kymmeniä kertoja (esimerkiksi tietyn mallin lukitseva kiusoitteluviesti, jonka sisältönä on pelkästään 160 pistettä).

Tuo puhelimen sekoittava viesti toimi muistaakseni Nokia 5110:lla. Sekään ei varsinaisesti ”lukinnut” puhelinta, vaan näytölle jäi joku ikuinen silmukka pyörimään, johon lienee auttanut puhelimen sammuttaminen virtanapista tai viimeistään akun irrottaminen.

Mainituista animaation kaltaisista kuvaviesteistä kirjassa ei sitten muuta mainitakaan. Huumorikiertoviestit toki mainitaan esimerkkeineen, ja mukana on klassikko:

*En ole yhtä lihaksikas kuin **Arnold Schwarzenegger** enkä yhtä karvainen kuin **Mel Gibson** enkä yhtä komea kuin **Brad Pitt**… mutta nuolen kuin Lassie.*

Toistaiseksi näyttää siis siltä, että akateemisessa tutkimuksessa aikalaisilmiö ei ole onnistunut jäämään haaviin.

Löytyipäs, kun vähän potki

Vaikka olin aikanaan googlaillut tuloksetta animaatiotekstiviestejä (viimeksi Skrollin 2019.3 kolumniin), yritin varmuuden vuoksi vielä kerran. Animaatiotekstiviesti, tekstiviestianimaatio ja sms-animaatio tuottivat taas laihan tuloksen, mutta kiitos yhdyssanavirheen, googlaus hakusanoilla tekstiviesti animaatio tuotti toivotun lopputuloksen: osoitteessa niko_85.tripod.com/tekstiviestit/animaatiot.htm on kokoelma animaatiotekstiviestejä. Sivulta löytyy myös linkki Microstella-yrityksen palveluun, jolla pystyi luomaan näitä animaatioita puhelimen omaa tekstiviestinsyöttöikkunaa näppärämmällä editorilla. Itse sivu ei ole enää olemassa, mutta Archive.orgin avulla voi tutkia sisältöä – johon kuuluu vielä kattavampi kokoelma alapäähuumorianimaatioita.

Apu-lehdessä 1.6.2001 julkaistu Sonera Zedin mainos soittoäänistä, logoista ja kuvaviesteistä. Käyttö lehdessä on sallittua tekijänoikeuslain 25 § nojalla.
Nokia 3310:n kuva Rainer Knäpper, Free Art License (artlibre.org/licence/lal/en).

Archive.orgin heikkous on kunnollisen vapaatekstihaun puute. Vanha nettisivu löytyy vain, jos tiedät osoitteen. Tämä tekee vanhojen verkkosivustojen etsimisestä hieman arkeologian henkistä salapoliisityötä. Tällä kertaa kävi tuuri, kun yksi sivu oli vielä verkossa Googlen ruokana.

Kyselin kokemuksia tekstiviestianimaatioista myös Skrollin irkkikanavalla. Moni muisti ilmiön heti ja muistutti myös tilattavista logoista ja soittoäänistä – joita onkin säilynyt hyvin koska niitä mainostettiin kuvineen aikakauslehdissä, ja löytyipä keskustelusta tällaisen palvelun (kismii.net) ylläpitäjäkin. Kansalliskirjaston digiarkiston haku ”logot ja soittoäänet” löytää runsaasti mainoksia joiden kultakausi on 2000-luvun alussa. ”Känny vireeseen!”, ”Lataa laatua!!”, ”Suosikit saattavat aiheuttaa riippuvuutta.” ja niin edelleen. Lopullisesti ilmiö hiipui 2010-luvun alussa, samalla kun Nokian puhelinten suosiokin.

Irkkikanavalla muisteltiin myös puhelimiin tilattavien juttujen laajaa brändäystä: yksi keskustelija kertoi olleensa töissä firmassa, joka möi hengellisiä soittoääniä ja kuvia ja raamatunlause päivässä -palvelua ihmisille. Kauppa kävi etenkin Pohjanmaalla. Toinen muisteli, että myös Voimalehdessä mainostettiin lukijakunnalle suunnattua puhelinsisältöä.

Teknisiä sivuhuomioita

Tekstiviestianimaatioiden heikkous oli, että ne toimivat kunnolla vain, jos lähettäjän puhelinmallissa oli samanlainen tai riittävän samanlainen näyttö kuin vastaanottajalla. Tiheämpi fontti sotki grafiikan, samoin liian pitkä rivi – kuten myös se, jos rivejä oli puhelimessa eri määrä. Alkuperäinen poroanimaatio toimi ainakin 2110:lla, ja muistaakseni siitä oli versio myös nuorison keskuudessa suositulle 5110:lle. Tämä ei ollut ongelma, koska vielä 1990-luvun lopulla puhelinmalleja oli kourallinen ja kaikki muistivat tasan tarkkaan, millainen puhelin kenelläkin oli.

On myös huomattava, että koko tekstiviestianimaatio oli ylipäätään keksintönä mahdollinen pienen suunnitteluteknisen valinnan takia: 1990-lukuiset nokialaiset vierittivät (tai ”vierittivät”) tekstiviestiä eteenpäin koko näytön kerrallaan, eivät rivi kerrallaan. Jälkimmäinen valinta olisi tehnyt animaatioiden tekemisen jos ei mahdottomaksi niin ainakin muutamaa kertaluokkaa haastavammaksi.

Lähteet

Esimerkkejä sisällöistä:
niko_85.tripod.com/tekstiviestit/animaatiot.htm
web.archive.org/web/20020602125815/http://www.mikrostella.com/
web.archive.org/web/20031012084401/http://kismii.net/

Elävältä museoon?

Olen ollut pitkään huolissani digitaalisen historian katoavaisuudesta. Peräänkuulutin taannoin (Skrolli 2019.3) Pelimuseon kaveriksi Meemimuseota, joka museoalan systemaattisuudella keräisi digitaalista kulttuuria talteen. Kaikki julkaistut kirjat ja äänitteet tallennetaan Kansalliskirjastoon, mutta esimerkiksi suositun Facebook-ryhmän sisällön tuhoamiseen riittää ylläpitäjän herkkä liipasinsormi. Mikään taho ei myöskään tallenna systemaattisesti internet-kulttuuria, vaikka esimerkiksi Kansalliskirjasto tekee arvokasta työtä tallentamalla jollain otannalla kotimaisia Twitter-keskusteluja ja Youtube-videoita.

Moni ”kokonaan netissä” tapahtuva asia myös jää internetiin, vaikka esimerkiksi Helsingin Sanomien Nyt-liite tekee satunnaisen säännöllisesti uutismaisia juttuja someilmiöistä. Pelkkä lehtien arkistointi ei siis tallenna riittävällä tasolla nettikulttuuria.

Perinteisessä mediassa – myös Nyt-liitteessä – netti-ilmiöiden käsittely jää myös valitettavan usein pinnalliseksi. Syykin tälle on selvä: laadukkaan jutun kirjoittaminen jostain tapahtumasta vaatii käytännössä, että on seurannut sitä alusta asti – muuten työmäärä venyy kohtuuttomaksi. Etenkin aikajanan hahmottaminen on haastavaa jo valmiiksi, ja oman lisämausteensa tuo materiaalin katoaminen ja sirpaloituminen.

Hyvä esimerkki on **Shia LaBeoufin** lipunryöstöepisodi, joka sivuutettiin kotimaisessa mediassa parilla lauseella. HS:n Nyt-liikkeen artikkelin mukaan ”Yhdysvalloissa alt-right-väki löysi salaiseen paikkaan pystytetyn He Will Not Divide Us -lipun, vaikka livelähetyksessä ei näkynyt kuin lippu ja plänttі taivasta. He paikallistivat sen muun muassa lentokoneiden reittejä tutkimalla.”

Kattavan reportaasin tapahtumaketjusta löytää Youtubesta **Internet Historian** -kanavalta, ja se on kieltämättä vaikuttava: taiteilijan piilottaman lipun suurpiirteinen sijainti etsittiin joukkoistetusti verkossa lentokoneiden reittejä ja tähtikuvioita tutkimalla. Koska lippua kuvaavassa videossa oli myös äänet, tarkka sijainti löytyi paikallisen asukkaan kautta, joka ajeli autolla ympäriinsä ja tööttäili samaan aikaan kun muut seurasivat videota. Varsin vaikuttava suoritus, joka ansaitsisi muutakin kuin pinnallisen maininnan. Syvällisyyttä kaipaisi myös sen pohtiminen, missä määrin kyse oli ”vaihtoehtoisesta oikeistosta” ja missä määrin tavallisista nuorista, jotka halusivat hyvät naurut.

Käsittelemättä jättäminen voi olla myös toimituksellinen valinta – kiusaajille ei haluta antaa huomiota. Mutta onko kyseessä kiusaaminen vai osa teosta?

Shia LaBeoufin tapauksen käsittely on helppoa: hän on julkisuuden henkilö omasta tahdostaan ja julkisuus – niin positiivinen kuin negatiivinenkin – sataa suoraan julkkiksen lompakkoon. Tekstiviestianimaatioon päätyneet Risto ja Seppo olivat pääuutislähetyksen aiheena, joskaan eivät omasta tahdostaan. Huumoritekstiviesti oli julkisuuden seuraus, ei syy.

Haastavaksi museointi ja myös uutisointi menee, kun tarinan päätähtenä on tavallinen ihminen, joka päätyy nettijulkkikseksi tahtomattaan tai yksittäisen töppäyksen seurauksena.

Internet-maailmassa kuka tahansa voi nousta nopealla aikataululla julkkikseksi yksittäisen töppäyksen tai ilkeämielisen ihmisen – yleensä molempien – seurauksena tahtomattaan. Suomessa tunnetuimpia esimerkkejä lienevät Risumies ja rähjäävä espoolainen ravintoloitsija. Molemmissa tapauksissa videon nettiin lataaja tuomittiin rikoksesta – Risumiehen tapauksessa vasta itse päähenkilön kuoleman jälkeen. Videot ovat vieläkin Youtubessa saatavilla.

Kuuluuko Risumies meemimuseoon, entä räyhäävä ravintoloitsija? Entä kuvalaudoilla levinnyt ”greikka-video” (sic), joka kuvasi nuorten suomalaisnaisten seksihurjastelua lomareissulla? Vuosikymmeniä sitten kuolleiden ihmisten elämän esittely ylä- ja alamäkineen museossa on sekä sosiaalisesti että juridisesti hyväksyttävää. Miten on elävien laita?

2000-luvun alussa avattiin eräs meemimuseon prototyyppi, **Marko H. Tammisen** ylläpitämä *Otaxin legendat* -verkkosivu, joka esittelee nimensä mukaisesti Teknillisen korkeakoulun ylioppilaskunnan uutisryhmissä vaikuttaneita sanataiteilijoita. Muutaman henkilön kohdalla lukee maininta, että kirjoitukset on poistettu asianosaisen pyynnöstä.

Näin on toimittu myös Meemi.infossa, jossa eräs aikansa nettijulkkis halusi poistaa artikkelista oikean nimensä. ”Totta kai poistamme nimet pyynnöstä, se on osa käytäntöjämme. Teitkin tämän itse, joka on ihan ok.”

Se on ok, mutta miten toimittaisiin ammattimaisesti ylläpidetyssä meemimuseossa? Anonymisoidaanko artikkelit jo valmiiksi? Vai jätetäänkö suosiolla muussa kuin toimitetussa mediassa käsitellyt netti-ilmiöt käsittelemättä? Vai laitetaanko nimen mainitsemiselle 50 vuoden embargo?

Kaikkea historiaa ei tallenneta systemaattisesti. Yksi konkreettinen esimerkki ovat vessanseinäkirjoitukset. Ne jäävät kyllä mieleen, mutta tiedossani ei ole tahoa, joka niitä keräisi systemaattisesti. Aina kun vessa maalataan uudestaan, kirjoitukset katoavat. Vessanseinäkirjoituksia pidettäneen niin arkisina, että kenellekään ei ole tullut mieleen niiden systemaattinen tallentaminen. Vaikka ehkä pitäisi – kulttuuriahan nekin ovat. Esimerkiksi tulivuorenpurkauksen hautaaman Pompeijin seinäkirjoitukset ovat tarkkaan dokumentoitu aineisto. Niistä on kirjoitettu suomenkielinen kirjakin, Ari Saastamoisen Räävitöntä! – Pompeijilaisia graffiteja. Kirjoitusten aiheet eivät ole juuri 2000 vuodessa muuttuneet.

Kun ILMAISPELIT valloittivat maailman

*Ilmaispelimalli on mullistanut pelialan. Peleistä, joita voi pelata ilmaiseksi, mutta jotka sisältävät vapaaehtoisia pelinsisäisiä maksuja, on kasvanut pelatuin digitaalisten pelien muoto ja kaikista tuottavin tapa tehdä peleillä rahaa. Samaan aikaan ilmaispelit ovat saaneet osakseen voimakasta kritiikkiä ja herättäneet vahvoja tunteita. Mistä tässä ristiriidassa on kyse? **Kati Alha** väitteli ilmaispelien vaikutuksista ja nostaa tässä tekstissä esiin väitöstyössä esiin nousseita näkökulmia.*

Teksti: Kati Alha Kuvankäsittely: Manu Pärssinen

Ilmaispelimalli nousi maailmanlaajuiseen suosioon erityisesti 2010-luvun vaihteessa, kun *FarmVillen* ja *Mafia Warsin* tapaiset Facebook-pelit keräsivät yllättäen satoja miljoonia pelaajia ja tekivät nopeita voittoja. Samaan aikaan PC-puolelta ilmaispelit nousivat aiempaa suositummaksi esimerkiksi *League of Legendsin* ja *World of Tanksin* julkaisun ja *Team Fortress 2:n* ilmaispeliksi muuttumisen kautta. Malli on sittemmin levinnyt monille alustoille ja monentyyppisiin peleihin, ja tällä hetkellä erityisen menestyneisiin peleihin kuuluvat esimerkiksi *Pokémon GO, Candy Crush Saga, Clash Royale, Fortnite, League of Legends* ja *Genshin Impact*. Joidenkin arvioiden mukaan ilmaispelit kattavat jo noin 80 prosenttia kaikista digipelien tuotoista. Taloudellinen merkitys on siis mittava, ja myös Suomen menestyksekäs peliala nojaa tällä hetkellä suurelta osin ilmaispelimalliin.

Kaikille ilmaispelien menestys ei ole ollut tervetullut, ja niistä on puhuttu pelien ja jopa koko pelialan pilaajana tai syöpänä. Ilmaispelit ovat herättäneet vastustusta, ilkkumista, halveksuntaa, jopa vihaa, ja niiden kehittäjiä on pidetty ahneina ja epämoraalisina.

Miksi pelit, joita kaikki voivat pelata ilmaiseksi, ovat herättäneet näin voimakkaita tunteita ja vastustusta? Väitöstyössäni keskityin erityisesti ilmaispeleihin liittyviin asenteisiin, kokemuksiin ja vaikutuksiin. Ilmaispeleihin liittyvät negatiiviset asenteet ovat kaksijakoiset. Yhtäältä ilmaispelit herättävät vastustusta, sillä ne eivät mahdu totuttuihin digipelien standardeihin siitä, millainen hyvä tai reilu pelikokemus on, ja toisaalta niissä nähdään eettisiä ongelmia, kuten kyseenalaisia tapoja saada pelaaja maksamaan.

Pelikokemuksen sabotointia

Pelikokemuksiin liittyvät ongelmat pohjautuvat erityisesti rahoitusmallin tuomiin haasteisiin. Siinä missä kertamaksullisten pelien suunnittelussa voi keskittyä luomaan parasta mahdollista pelikokemusta, ilmaispelien tulee onnistua luomaan pelistä sekä tarpeeksi hyvä että sellainen, että se houkuttelisi tarpeeksi monen maksamaan pelaamisen aikana. Siten peleihin on luotu esimerkiksi erilaisia pelikokemusta hankaloittavia tai jarruttelevia mekaniikkoja, joita voi kiertää maksamalla. Tämä voi tarkoittaa vaikkapa sitä, että peli rajoittaa peliaikaa odotusmekaniikoilla tai lisää siihen erityisen vaikeita kohtia tai toisinaan helppoja mutta itseään toistavia, tylsiäkin tehtäviä. Rahaa käyttämällä voi sujuvoittaa tai helpottaa pelaamista tai ohittaa pelin sisältöä.

Maksamisen tuominen osaksi pelikokemusta ja muuhun kuin parhaaseen pelikokemukseen tähtääminen voi tuntua väärältä ja ilmaispelien mukanaan tuomalta uudelta tavalta suunnitella pelejä. Ihan näin ei kuitenkaan ole. Ilmaispelien suunnittelulogiikkaa lähellä ovat esimerkiksi jo digipelien alkuaikojen arcade-hallien kolikkopelit, joiden suunnittelu perustuu osin juurikin hankaloittaviin mekaniikkoihin. Koska pelaajan ei haluttu pelaavan liian kauaa yhden kolikon varassa, peleihin lisättiin aikarajoitteita tai jatkuvasti vaikenevia tasoja, jolloin pelaaja saattoi maksaa lisää pelisession aikana voidakseen jatkaa pelaamistaan. Kolikkopelejä kohtaan ei kuitenkaan syntynyt samankaltaista vastustusta, ehkä osin koska ne olivat mukana muodostamassa digipelialaa hyvin varhaisessa vaiheessa, jolloin ”oikeanlainen” pelikokemus ei ollut vielä sementoitunut peliharrastajien parissa. Kolikkopeleissä kertamaksut myös pysyvät pieninä, kun ilmaispeleissä suuriakin summia voi tuhlata hetkessä, jolloin pelinkehittäjän näkeminen ahneena rahastajana on huomattavasti helpompaa.

Kiinnostavaa hankaloittavissa mekaniikoissa on se, etteivät ne aina suinkaan huononna pelikokemusta – ainakaan kaikille. Esimerkiksi odotusmekaniikkoja käyttävä peli voi luoda rytmiä pelaamiselle ja rajoittaa pelaamista positiivisella tavalla, kun taas haasteeseen pohjautuvat maksut voivat luoda uuden metahaasteen, jossa pelataan maksuja vastaan. Ilmaispelit ovat haasteidensa kautta luoneet omanlaista pelaamista, jota moni näiltä peleiltä jo odottaa ja toivoo.

Paha, pahempi, pay-to-win

Aina ilmaispeli ei yritä rahastaa hankaloittamalla peliä, vaan se voi tarjota myös toisenlaisia motivaatioita maksaa. Usein kaikista kyseenalaisimmaksi maksetuksi sisällöksi koetaan sellainen sisältö, joka parantaa pelisuoritusta muihin pelaajiin nähden. Tämä voi tarkoittaa nopeampaa edistymistä pelissä, jolloin pelaajasta tulee epäsuorasti voimakkaampi ilmaispelaajia nopeammin, tai tarjota voimaa suoraan maksavalle pelaajalle vaikkapa parempien maksullisten varusteiden kautta. Pelejä, jotka merkittävästi muuttavat maksavien ja ilmaiseksi pelaavien pelaajien tasapainoa kilpailullisissa peleissä, kutsutaan pay-to-wineli maksamalla voittoon -peleiksi.

Rahalla pärjääminen nostaa esiin erityisesti reiluuden käsitykset sekä kiinnostavia näkemyksiä rahan ja ajan suhteesta. Pelaajien keskuudessa on jyrkkiäkin näkemyksiä siitä, ettei rahalla tule edetä peleissä missään muodossa, vaan ainoastaan omalla ajankäytöllä ja taidoilla tulisi olla väliä. Tämä pohjaa meritokraattiseen ajatteluun, jonka mukaan kaikilla on yhtäläinen mahdollisuus pärjätä riippumatta siitä, kuinka paljon rahaa heillä on käytettävissään. Harvempi peli on kuitenkaan täysin reilu tässä mielessä, sillä pelaamiseen liittyy monia pelaajia eri asemiin asettavia tekijöitä, kuten kalliiden laitteiden antamat edut tai ei-vammaisten suosiminen, toksiset, erityisesti vähemmistöjä syrjivät yhteisöt sekä taitojen siirtymistä pelien välillä suosivat konventiot. Toisaalta myös käsitykset reiluudesta ja rahasta vaihtelevat kulttuurista toiseen, ja esimerkiksi länsimaisten ja aasialaisten pelaajien keskimääräiset asenteet rahan avulla voittamisesta eroavat tutkimusten mukaan toisistaan merkittävästi.

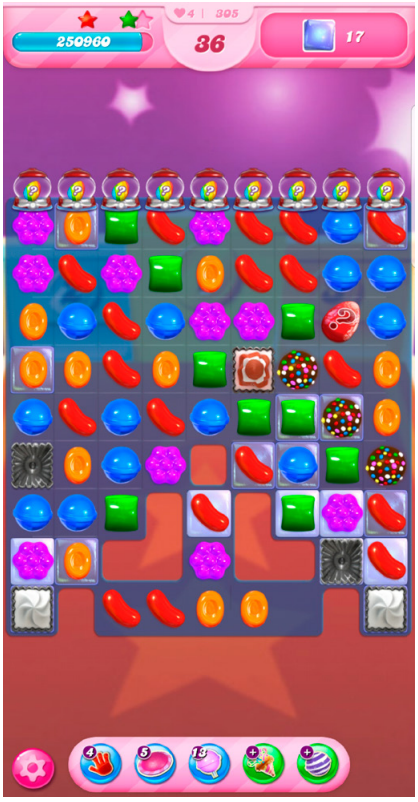
Suoran voiman ostamiseen nähden täysin toisessa päässä on kosmeettinen sisältö, eli sellainen sisältö, joka ei vaikuta pelin toimintaan vaan muuttaa sitä esimerkiksi visuaalisesti. Tyypillinen kosmeettinen sisältö voi olla uusia skinejä eli ulkonäköjä hahmoille tai varusteille, tai uusia eleitä kuten tansseja pelihahmoille. Ilmaispelikritiikissä yksi esiin tullut mielipide onkin, että hyvien ilmaispelien tulisi myydä vain tämän kaltaista sisältöä, eikä mitään, mikä muuttaa pelin toimintaa. Kosmeettiseen sisältöön keskittyminen ei kuitenkaan toimi kaikissa pelityypeissä, vaan yleensä lähinnä sellaisissa moninpeleissä, joissa muiden pelaajien hahmot ovat näkyvässä roolissa. Niinpä kun esimerkiksi *Fortniten* kaltainen monen pelaajan toimintapeli on menestynyt nimenomaan kosmeettisella sisällöllä, on vaikea kuvitella vaikkapa *Candy Crush Sagan* tapaisen pulmapelin pääsevän samaan myymällä pelkästään erilaisia ulkonäköjä, animaatioita ja ääniä.

Ilmaispelien murros

Ilmaispeleihin liittyvät negatiiviset asenteet ovat jossain määrin lieventyneet vuosien saatossa, osin ehkä kehittyneiden, ”hyvien” ilmaispelien, osin tottumisen ja uusien pelaajasukupolvien kautta. Samalla kun yhä useampi viihtyy ilmaispelien parissa, niiden sisällä on edelleen selkeää jakoa asenteissa. Peli, joka pystyy täyttämään perinteisemmät arvot hyvästä pelistä eikä myy pelin kulkuun vaikuttavaa sisältöä, voidaan nähdä ”oikeaksi” peliksi, tai ainakin hyvin tehdyksi ilmaispeliksi. Tähän rinnastuvat esimerkiksi useat mobiilipelit, joita pidetään vähempiarvoisina tai ongelmallisempina. Samalla mobiilipelit tuottavat selvästi suurimman osan ilmaispelien tuotoista.

Tälle ilmaispeleihin yleisemmin sekä mobiili-ilmaispeleihin erityisemmin liittyvälle epäsuhdalle voi päätellä helposti kaksi selitystä: joko ongelmallisia ja vihattuja pelejä pelataan ja niistä maksetaan vastentahtoisesti, mahdollisesti pelien manipuloivien mekaniikkojen vuoksi, tai pelejä kritisoi pieni äänekäs joukko samalla, kun suuret pelaajakunnat pelaavat niitä ja maksavat niistä mielellään. Todennäköisesti totuus ei ole yksiselitteisesti kumpikaan näistä vaihtoehdoista, mutta on oletettavaa, että pelikulttuurissamme nousevat esiin voimakkaimmin tietyntyyppiset äänet samalla, kun pelaajakunta on paljon laajempi ja mieltymyksiltään monimuotoisempi.

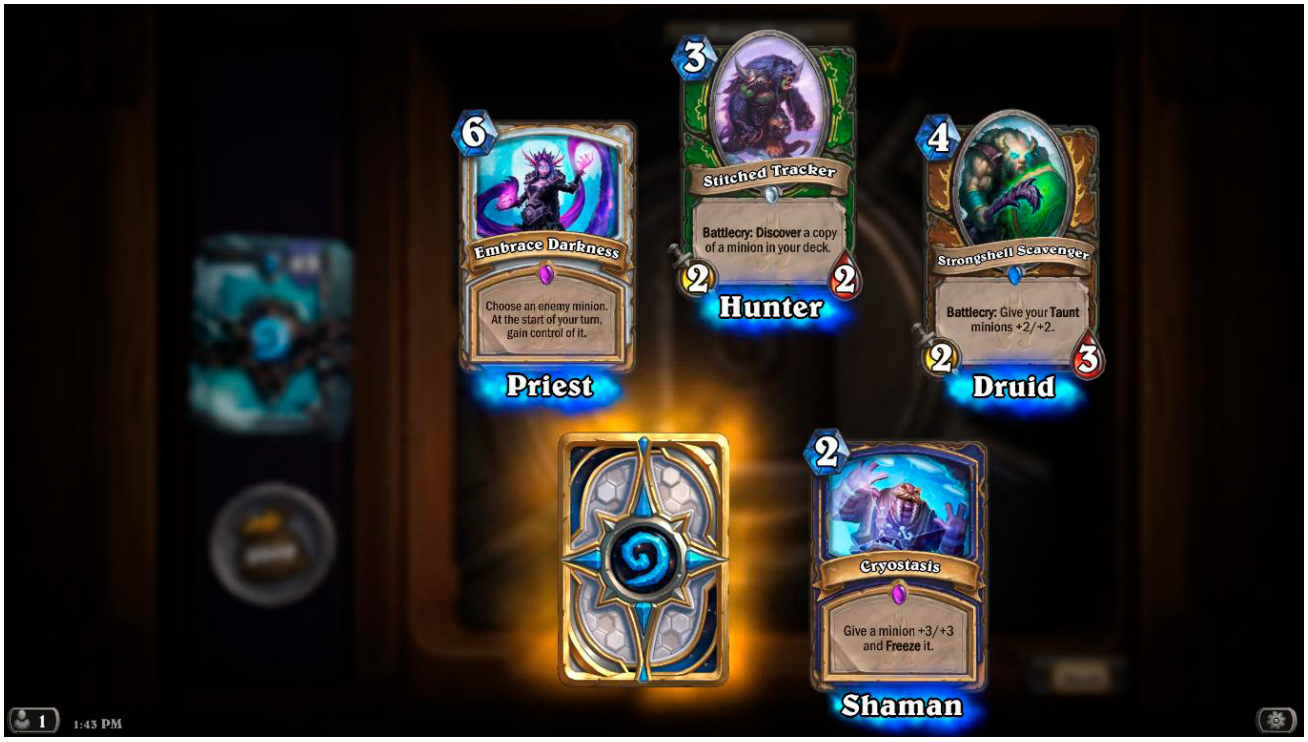
Myös maksamisesta ilmaispeleissä on tullut normaali tapa. Esimerkiksi *FarmVillen* aikoihin oli tutkimuksissakin hankala tavoittaa niitä pelaajia, jotka peleistä maksoivat. Malli perustui siihen, että vain pieni, muutaman prosentin joukko maksoi lainkaan, ja maksavista pelaajista vain pieni joukko maksoi niin paljon, että he kannattelivat koko peliä. Sittemmin yhä useampi peli on päässyt tässä parempaan tasapainoon: monissa hyvin pärjäävissä peleissä maksava joukko voidaan usein laskea kymmenissä prosenteissa. Tämä malli on terveempi niin pelifirman kuin pelaajien kannalta: toisaalta rahantulo ei ole riippuvainen jopa yksittäisistä suurkuluttajista, jolloin tilanne on vakaampi kun pelaajakunta vaihtuu, ja toisaalta pelaajia ei enää yleensä yhtä aggressiivisesti painosteta maksamaan yhä suurempia summia. Maksukattojen puute aiheuttaa silti yhä ongelmia: jos peleihin saa hetkessä kulutettua tuhansia euroja, on edessä väkisinkin pohdintaa siitä, mitä eettisiä ongelmia maksuihin voi liittyä.



Jo kymmenen vuotta sitten julkaistu Candy Crush Saga on edelleen yksi menestyneimmistä ja tunnetuimmista ilmaispeleistä.

Monimuotoisia eettisiä haasteita

Monet hyvään pelikokemukseen ja reiluuteen liittyvät asenteet ovat usein subjektiivisia, eikä niihin ole suoraan oikeita tai vääriä vastauksia. Sen sijaan ilmaispeleihin liittyvät eettiset ongel-



Hearthstonessa pelaaja voi hankkia uusia kortteja ostamalla korttipakkoja, jotka sisältävät viisi satunnaista korttia. Hyvällä onnella paketista löytyy harvinaisuuksia.

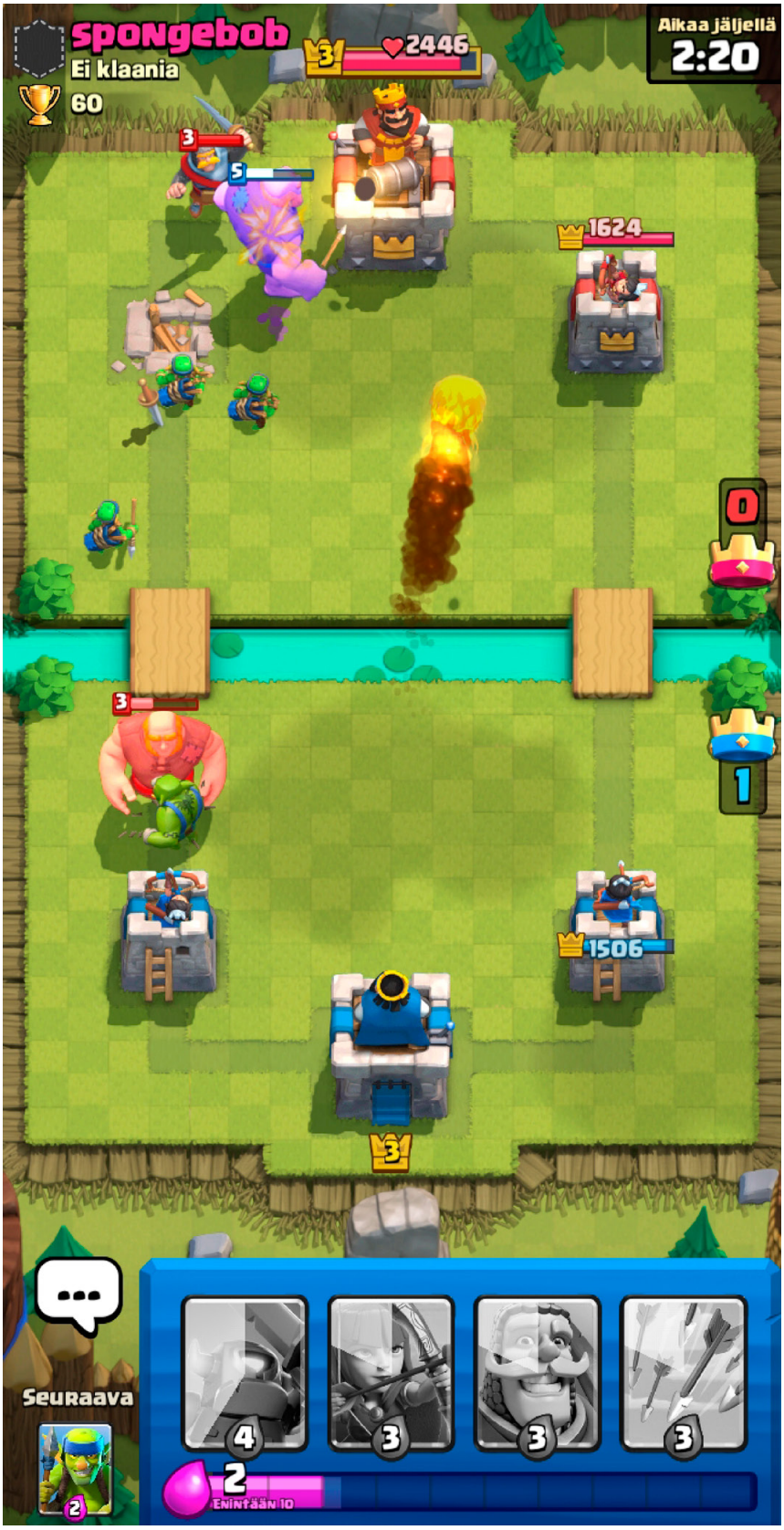
mat vaativat vakavampaa tarkastelua ja mahdollisesti niihin puuttumista. Ilmaispelien eettiset ongelmat kohdistuvat muun muassa läpinäkyvyyteen, ongelmapelaamiseen, alaikäisille markkinointiin ja samankaltaisuuteen uhkapelien kanssa.

Läpinäkyvyyden puute voi tarkoittaa sitä, ettei pelaajalle anneta tarpeeksi tietoa mahdollisista pelin sisältämistä maksuista tai niiden todellisesta hinnasta. Hyvä esimerkki tästä on se, että ilmaispelejä markkinoitiin aiemmin laajasti ilmaisina, mikä antoi kuluttajille väärän kuvan pelien maksuista. Tätä on sittemmin kontrolloitu siten, että ilmaispelien tulee sisältää kuvauksessaan maininta pelinsisäisistä ostoista. Pelinsisäisten ostojen varsinaiset hinnat saattavat hämärtyä, jos pelaaja ei voi ostaa esineitä suoraan rahalla vaan ostamalla ensin pelinsisäistä valuuttaa. Osa ongelmista saattaa liittyä jopa harhaan johtamiseen, kun peliä vaikkapa mainostetaan tavoilla, jotka antavat väärän kuvan varsinaisesta pelistä.

Ongelmapelaamiseen liittyvät eettiset ongelmat ovat monimuotoisia ja haastavia. Tällöin pelaaja pelaa siten, että se vaikuttaa negatiivisesti muuhun elämään, kuten yli varojen pelatessa. Tähän voi liittyä esimerkiksi se, onko peli erityisen addiktoiva tai manipuloiko se pelaajia maksamaan sisällöistä. Peli voi myös käyttää erityisen aggressiivisia keinoja yrittäessään saada pelaajaa maksamaan tai pyrkiä saamaan lisämaksuja niiltä pelaajilta, jotka ovat jo maksaneet enemmän. Vielä kyseenalaisemmalle alueelle mennään, kun pelejä pelaavat lapset ja nuoret, ja samoja markkinointikeinoja käytetään heihin. Joissain tapauksissa jopa selkeästi lapsille suunnatuissa ilmaispeleissä on käytetty aggressiivista markkinointia. Näitä ongelmia on käsitelty laajasti, mutta vaikka samanlaisia ylilyöntejä ei enää yhtä usein nähdä, on esimerkiksi keskustelu ilmaispelien ikärajoista ja muista rajoituksista nuorille pelaajille edelleen paikallaan.

Erityisen ongelmallisen rahankäytön suhteen ilmaispeleistä tekee mainittu maksukattojen puute: niissä ei monien muiden pelien tapaan ole rajoituksia sen suhteen, kuinka paljon yksittäinen pelaaja voi käyttää rahaa peliin. Niinpä ongelmapelaamisella voi olla merkittäviä taloudellisia vaikutuksia. Koska ne tässä suhteessa ovat rahapelien kaltaisia, on myös ehdotettu, että ne voisivat käyttää rahapeleistä tuttuja ratkaisuita. Tällaisia voivat olla esimerkiksi kulutuskatot ja seurantatyökalut omasta rahan tai myös ajan käytöstä. Näitä työkaluja ei kovin moni ilmaispeli ole ottanut käyttöön, osin ehkä koska niiden pelätään pienentävän pelin tuottoja, sillä monien ilmaispelien supertuotot edelleen perustuvat siihen, että pelatessa maksetaan toisinaan hyvinkin paljon. Kun seuranta ja kulutuskatot ovat kunkin pelaajan määriteltävissä, toisivat ne lisää valtaa juuri ongelmapelaamisen omaehtoiseen hallintaan ja siten tekisivät peleistä vastuullisempia. Tällöin ne pelaajat, joilla on varaa ja halua maksaa isoja summia, voisivat edelleen tehdä niin.

Tällä hetkellä näkyvin kriittinen keskustelu ja tutkimus ilmaispeleihin ja potentiaaliseen ongelmapelaamiseen liittyen koskee niiden uhkapelimäisyyttä, erityisesti yllätyslaatikoiden eli loot boxien muodossa. Yllätyslaatikot sisältävät sekä harvinaista arvokasta tavaraa että huomattavasti yleisempiä arvottomampia esineitä, joten jokainen yllätyslaatikko muistuttaa arpaa. Ongelmia tuleekin, jos ilmaispelimekaniikka nähdään uhkapelaamisen muotona, mutta peli ei ole rahapelilainsäädännön alainen. Tällä hetkellä monissa lainsäädännöissä pohditaankin juuri yllätyslaatikoiden kohtaloa, ja joissain yksittäisissä maissa ne ovat tällä hetkellä kokonaan kiellettyjä. Jos tämänkaltaisilta kokonaisvaltaisilta kielloilta halutaan välttyä, olisi tärkeää, että pelifirmat itse kehittäisivät reiluja ja eettisiä ratkaisuja ongelmiin.



Suomalaisen Supercellin kehittämä Clash Royale on yhdistelmä tornipuolustusta, taisteluareenamoninpeliä ja keräilykorttipeliä.

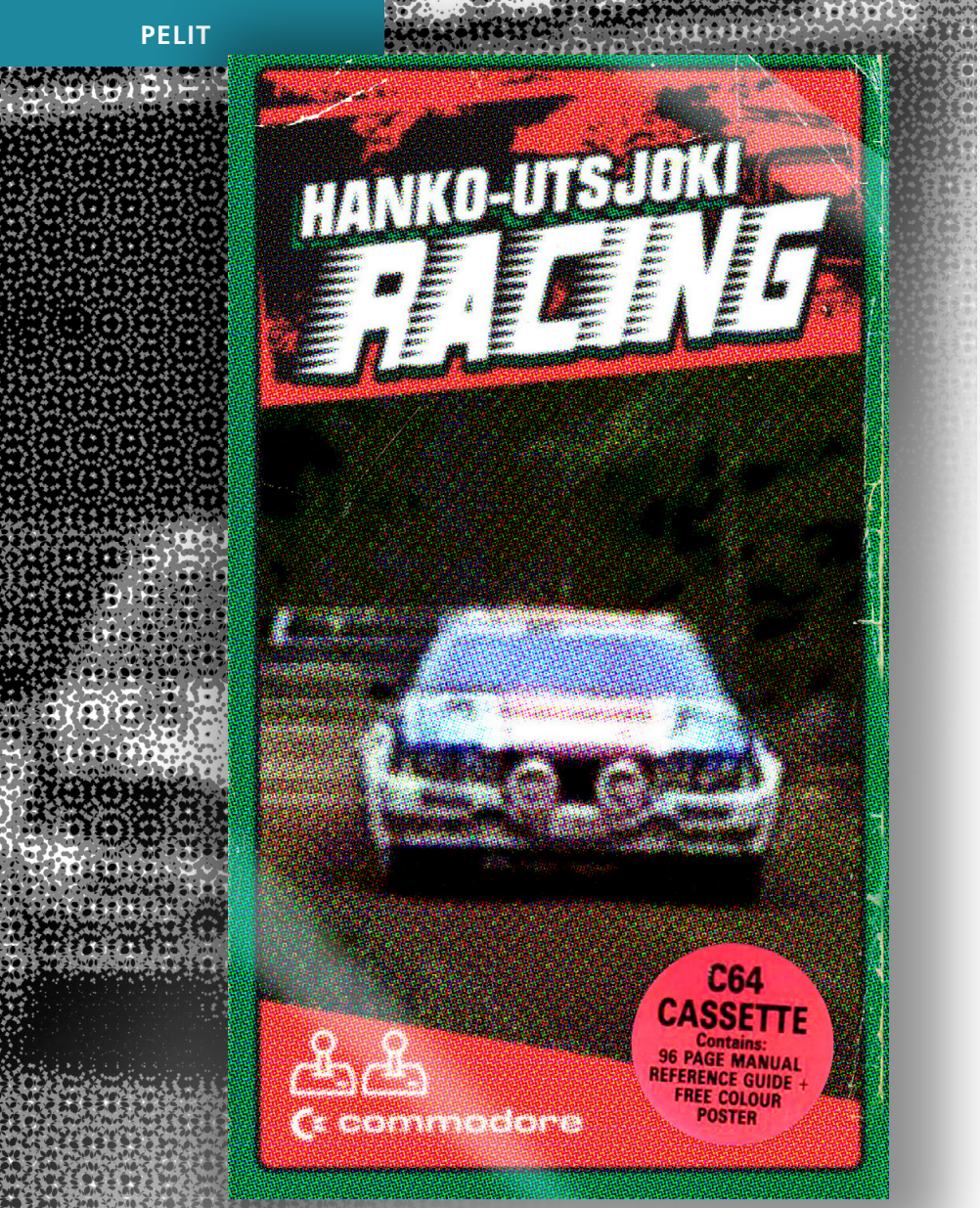
Kenen mielipiteellä on väliä?

Osin näennäisen vahva negatiivinen suhtautuminen ilmaispeleihin johtuu siitä, keitä pelaamista koskevissa keskusteluissa kuullaan. Näkyvä pelikulttuuri keskittyy pitkälti perinteisempien digipelien peliharrastajiin, ja ilmaispelit on ehkä osin koettu uhkana tietynlaiselle ideaalille pelikokemuksissa. Myös tutkimuksiin vastaavat helpommin sellaiset henkilöt, joilla on voimakkaita näkemyksiä tutkittavaan asiaan, suuntaan tai toiseen. Siten on todennäköistä, että esimerkiksi ne ilmaispelien pelaajat, jotka eivät koe pelejä tunteita herättävinä tai välttämättä erityisen merkityksellisinäkään, jäävät helposti tutkimuksissa ja keskusteluissa kuulematta. Esimerkiksi yhdessä omista ilmaispeleihin liittyvistä kyselyistämme, johon saimme kyllä hyvin vastauksia, oli vastaajajoukko tietyiltä osin selkeästi rajoittunut. Vaikka kyselyyn sai vastata kuka tahansa, reilu 90 prosenttia vastaajista oli miehiä ja lähes 95 prosenttia alle 40-vuotiaita. Siten tutkimuksemme, joka kartoitti ilmaispelien ostojen motivaatioita, kuvaa vain tietyn pelaajakunnan näkemyksiä. Tutkimuksessa tuleekin tehdä lisätyötä sen eteen, että tarkastelussa kuullaan monenlaisia pelaajia, sillä kaikkien pelaajien äänellä on väliä.

Pelit eivät ole saareke, jota voi täysin suojella tai pitää hallussa. Pelit ovat valtava, hajanainen kokonaisuus, jota ei voi tai tulisikaan vaatia tietyn ryhmän omistamaksi. Pelit kuuluvat kaikille, ja niistä myös riittää kaikille. Tällä hetkellä pelit tarjoavat kokemuksia laajemmalle yleisölle kuin koskaan, ja hyvä niin. Ilmaispeleillä on ollut tässä leviämisessä tärkeä rooli. Tämä ei kuitenkaan tarkoita sitä, että mitä tahansa tulisi sallia. Samalla kun pelaajat itse toiminnallaan ratkaisevat osaltaan sitä, millaiset pelit menestyvät, tulee epäreiluja tai epäeettisiä käytäntöjä tarkastella kriittisesti ja tarvittaessa rajoittaa lainsäädännöllä. Ilmaispelit tulee ottaa vakavasti, ja niitä voi omista pelimieltymyksistä huolimatta arvostaa – myös tutkimalla niitä avoimesti ja kriittisesti.

On vaikea ennustaa, millainen ilmaispelien tulevaisuus tai rooli on. Kriittisenä seikkana voi nähdä ainakin lainsäädännön ja ulkoa tulevat rajoitukset, ja miten ilmaispelit kokonaisuudessaan kehittyvät vastuullisemmiksi. Ilmaispelit kehittyvät myös toisin tavoin, ja uudet myytävät sisällöt ja mekaniikat korvaavat toisia. Toistaiseksi ilmaispelit ovat jatkaneet vuosittaista kasvuaan, mutta on todennäköistä, että tämäkin kasvu tasaantuu jossain vaiheessa. Yksi asia on kuitenkin varma: piti niistä tai ei, ilmaispelit eivät ole menossa mihinkään.

Alhan väitöskirja ”The Rise of Free-to-Play: How the Revenue Model Changed Games and Playing” on luettavissa osoitteesta urn.fi/URN:ISBN:978-952-03-1774-4



EI NÄIN

Kyltymätön virtuaalikuski

Haluan elämänmakuisen autosimulaattorin, jossa on valtava kartta ja riittävän tylsää.

Teksti: Mikko Heinonen Kuvankäsittely: Manu Pärssinen

Olen aina rakastanut videopelejä – ja rakastan edelleen. Aivan erityisesti olen rakastanut autopelejä, sillä autolla ajamisen simulointi tuntui pienestä bensalenkkarista joskus jopa päällimmäiseltä syyltä pelaamiseen. Pelin ruudulla saatoin ajaa ”Ferrarilla”, vaikka se oli vain pieni röpöinen sprite, joka väisteli muita röpöjä ankeasti välkkyvällä taustalla, jonka piti kuvata alla vierivää moottoritietä. Kolikkopelin loistosta oltiin kaukana, mutta eipä kolikkopeliin olisi ollut varaakaan.

Kun ensimmäiset kolmiulotteista grafiikkaa käyttävät autopelit ilmestyivät, olin aivan haltioissani. Pelasin jopa perin keskinkertaista *Beverly Hills Copia*, koska siinä oli siedettävä autoiluosuus. Rakastin *Hard Drivin'* -kolikkopeliä, mutta en tarpeeksi, että olisin kestänyt sen hirveää kotikoneversiota. *Stunts* olikin sitten jo aivan eri maata ja varmaankin yhä yksi eniten pelaamiani pelejä.

Ala-asteikäisenä vilkas mielikuvitukseni puuhasteli usein pelikonseptien parissa. Ohjelmoinnista en ymmärtänyt juuri mitään, mutta visioita minulla on aina ollut sitäkin enemmän. Kehittelin itsenäisesti muun muassa konseptin, joka perustui tupakkamerkin sponsoroimaan Camel Trophy -maastoautokilpailuun. Ideana oli siis ajaa jeepeillä pitkin viidakkopolkuja ja suorittaa välillä erilaisia tehtäviä, kuten rakentaa siltoja, korjata autoa ja niin edelleen. Muistan selittäneeni ajatusta isälleni niin laajasti yhdellä kävelylenkillä, että hän kysyi erikseen, onko tämä nyt jo jokin oikea peli vai vain oma ideani.

Myöhemmin sain tietää, että peli nimeltä *Camel Trophy* oli olemassa ja että se oli aika huono. Sitten löysin Epyxin julkaiseman pelin nimeltä *4x4 Off-Road Racing* (1988), joka vaikutti olevan kuin minulle tehty. Ostin sen Amigalle joululahjarahoillani ja petyin katkerasti, kun lupaavan alun jälkeen itse ajo-osuus olikin surkea *Buggy Boy* -kopio.

Magnum Opus

Tämä maastoajopeli ei kuitenkaan ollut omasta mielestäni edes kuumottavin ajatus. Se oli *Hanko–Utsjoki Racing*, jonka kehittelimme kaverini **Jussin** kanssa pihalla leikkiessämme. Konseptina siis oli ajaa Suomen päästä päähän, normaalin liikenteen seassa, mahdollisimman nopeasti. Painopiste oli realismissa, eli siinä että auton polttoaine kuluu eikä sillä voi kolaroida järjettömästi, ja että ajaminen kestää niin kauan kuin se oikeastikin kestäisi. Matkalla pitäisi olla oikeita maamerkkejä. Käsitimme toki, että silloisten kotitietokoneiden resursseilla ajatuksen toteuttaminen saattaisi olla kauniisti sanottuna haastavaa, mutta saahan sitä unelmoida.

Visiointi oli kuitenkin vain puoli-leikillistä, sillä näimme tekniikan kehittyvän harppauksin ympärillämme. Siinä, missä VIC-20:llä autopelit olivat usein ylhäältä kuvattuja ja yhdellä ruudulla, C64:n aikakausi toi jo mukanaan kuvakulman auton takaa. ZX Spectrum -peli *Turbo Esprit* osin vektorigraafisine näkymineen näytti lehdissä erinomaisen kuumottavalta, mutta kaupassa en sitä koskaan nähnyt, joten hankkimatta jäi. Myöhemmin kuulin myös, että C64-käännös oli tunaroitu. Toinen kiinnostava olisi ollut Activisionin *The Great American*

Cross-Country Road Race (1985), josta en tuolloin tiennyt mitään. Itse asiassa se oli melko lähellä tätä perusideaa, joskin hyvin primitiivisesti toteutettuna. Kyseessä kun oli paranneltu versio Atari 2600:n pelistä *Enduro*.

Lopulta en ollut edes niin kiinnostunut toteutustavasta kuin vapaudesta. Halusin tehdä enemmän kuin vain ajaa kilpaa ennalta määritellyllä radalla, etenkin valita itse reittini ja vastata valintojeni seurauksista. Jotain tällaista koin Radarsoftin *Liikennepelissä* (1985), jota pelasin suhteettoman paljon, mutta se oli sivulta kuvattu, muuten perin yksinkertainen ja sen latautuminen kasetilta kesti aivan tolkuttoman kauan.

Koeajolta toiselle

Ihastuin *Test Driveen* (1987) ensisilmäyksellä, koska autoa pääsi ohjaamaan sisältä ja ajamaan normaalin liikenteen seassa. Opettelin peliä ajamalla ensin nopeusrajoitusten mukaisesti, jotta auto pysyi hallinnassa eikä poliiseista ollut haittaa. Se oli lopulta yllättävän mukavaa. Peli oli kuitenkin lineaarinen eikä mahdollistanut isompaa tutkimista. *Test Drive II: The Duel* sentään tarjosi erilaisia maisemia lisälevykkeillä.

Huimin kokemus oli kuitenkin vektorigrafiikkaa käyttävä *Test Drive III* (1990), joka oli eri kehitystiimin käsialaa. Se oli itse asiassa ajettavuudeltaan kohtalaisen surkea autopeli, varsinkin Stuntsiin verrattuna, mutta kartan avoimuus ja vapaus määrittää reitit tekivät syvän vaikutuksen. Kisan voittaminen ei niinkään kiinnostanut, kunhan ajelin huvikseni. Hankin jopa *Road & Car* -lisäosan, jolla sai lisäkartan ja sitä kautta paljon enemmän tutkittavaa. Silti olisin toivonut, että ajaminen olisi tuntunut mukavammalta. Spectrum Holobyten lupaava *Vette!* kaatui samaan vähän aiemmin. Ajatus kaahailusta Corvettella pitkin San Franciscon katuja oli sinällään kiinnostava, mutta itse peli tönkkö ja vähän turhan ruma low poly -malleineen.

Vuosituhannen vaihteessa Microsoft toi Windowsille useita 3D-kiihdytystä käyttäviä pelejä, joista yksi oli *Midtown Madness*. Vaikka sekään ei ollut kaipaamani massiivisen skaalan vapaa autoilu, kohtuullisen suuressa ja elävässä kaupungissa huristelu tuotti kuitenkin suurta hupia. Pian sitten ilmestyikin *Grand Theft Auto III* jatko-osineen, joiden parissa unohdin hetkeksi alkuperäisen haaveeni ja tyydyin sandbox-ajeluun.

No joko nyt?

Vuonna 2006 julkaistiin *Test Drive Unlimited*, josta innostuin välittömästi. Peli sijoittui Havaijin saarelle, jossa oli tilaa kaahailla unelma-autoilla. Teknisesti se oli lievästi sanoen kunnianhimoinen ja varsinkin moninpeli jumitteli tuon tuostakin, mutta pidin siitä silti yllättävän paljon. Valitettavasti *TDU2* oli kaikin puolin kökömpi esitys ja ennen kaikkea jakoi ajettavan alueen kahteen eri lataukseen.

Saman idean toteutti paljon paremmin Microsoftin *Forza Horizon* (2012), tosin ei vielä ensimmäisessä iteraatiossaan. Xbox 360:n teho ei riittänyt aidosti avoimeen karttaan, mutta Xbox Onelle ilmestynyt kakkososa tarjosi oikeasti melkoisen ison ja avoimen alueen tutkittavaksi. Vitososaan asti ehtinyt sarja on hiljalleen parantunut jokaisen osansa myötä, mutta omaan lapsuuden visiooni nähden siinäkin on kuitenkin omat puutteensa.

Niin TDU:ssa kuin Forzassa meininki on nimittäin *liian hienoa*. Ajettavana on upeita mutkaisia teitä ja pitkiä suoria, joilla voi kaahata miljoonien hintaisilla urheiluautoilla. Itse ajattelin, että täysin avoimessa maailmassa on myös kapeita, kuraisia kinttupolkuja ja vuoden 1987 Toyota Corolla. *My Summer Car* ja sen elämäsimulaatio kutittelee kuitenkin vähän eri hermoa.

Vuonna 2014 Ubisoft julkaisi *The Crew'n*, jossa mainostettiin olevan käytettävissä koko Yhdysvaltain kartta. Olin kiinnostunut mutta skeptinen asian suhteen, sillä pettymyksiä oli vuosien mittaan tullut lukuisia. Yllättävää kyllä, kartta oli oikeasti melkoisen suuri, jos nyt kohta ei kovin realistinen. Valitettavasti sen päälle rakennettu peli ei jaksanut kiinnostaa kovin pitkään, ja jälleen järjetön kaahailu oli niin helppoa, että etäisyydet kutistuivat kummasti. Jatko-osa *The Crew 2* saa edelleen lisäsisältöä, mutta omaan makuuni sen toiminta on liian pintaliitoa.

Tylsyyden suloinen syli

Muutama vuosi sitten sain kuitenkin kokea ainakin yhden pelin parissa juuri sellaisia tunteita, joita *Hanko–Utsjoki Racing* -konseptikin aikoinaan herätti. Se peli oli *American Truck Simulator* (2016). Rahdin kuljettaminen kaupungista toiseen kaikessa rauhassa on rentouttavaa, ajoittain jopa tylsää, ja juuri se siinä myös viehättää. Pelissä on myös mahdollista tehdä erilaisia virheitä, joista ajaminen keskeytyy. Pitemmän päälle pelistä nauttimisen esti kuitenkin rahdin kanssa kikkailu ja yhdistelmien peruuttelu, josta en erityisemmin perustanut.

Osin samasta syystä hankin jo aikoinaan *Microsoft Flight Simulatorin* MS-DOSille. Halusin kokeilla, miltä tuntuu oikeasti lentää pienkoneella kaupungista toiseen. Ajatus tuntui hienolta, mutta valitettavasti maaston yksityiskohtien täydellinen puuttuminen matkalta teki puuhasta uuvuttavampaa kuin mihin olin tuolloin valmis, joten tyydyin lentelemään melko lyhyitä välejä. Nykyisin MSFS sisältääkin sitten valtavan kartta-aineiston ja tuhottomasti yksityiskohtia – mutta en oikeastaan halua lentää, vaan ajaa paikasta toiseen.

Tässä onkin olennainen ero peligenrejen välillä. Lentosimulaattorissa on oleellista, miltä maasto näyttää ilmasta katsoen, mutta valtavan kartan avoimen maailman autopelissä sen pitäisi olla ajettavissa ja jotenkin generoitua karttadatasta. Tämä on kertaluokkainen tekninen haaste, joten en edelleenkään pidätä hengitystäni sen suhteen, että voisin ajella realistisissa ympäristöissä missä päin maailmaa haluaisin.

Jotakin hyvin samaa kuin omissa ajatuksissani on myös indie-pelissä *The Long Drive* (2019). Dystooppiseen, autioon maailmaan sijoittuvassa pelissä ei ole muuta ideaa kuin ajaa mahdollisimman pitkälle proseduraalisesti luodussa maailmassa, etsien autoon varaosia ja polttoainetta ja varoen verenhimoisia pupuja. Kilometrit kertyvät oikeaa tahtia, polttoaineen kulutus on samoin realistinen. Auto myös kuluttaa renkaita, öljyä ja jäähdytysnestettä. Vaikka peli on edelleen valitettavan buginen, muuta liikennettä ei ole ja ajomatka keskeytyy usein kuljettajasta riippumattomasta syystä, olen pelannut sitä kymmeniä tunteja.

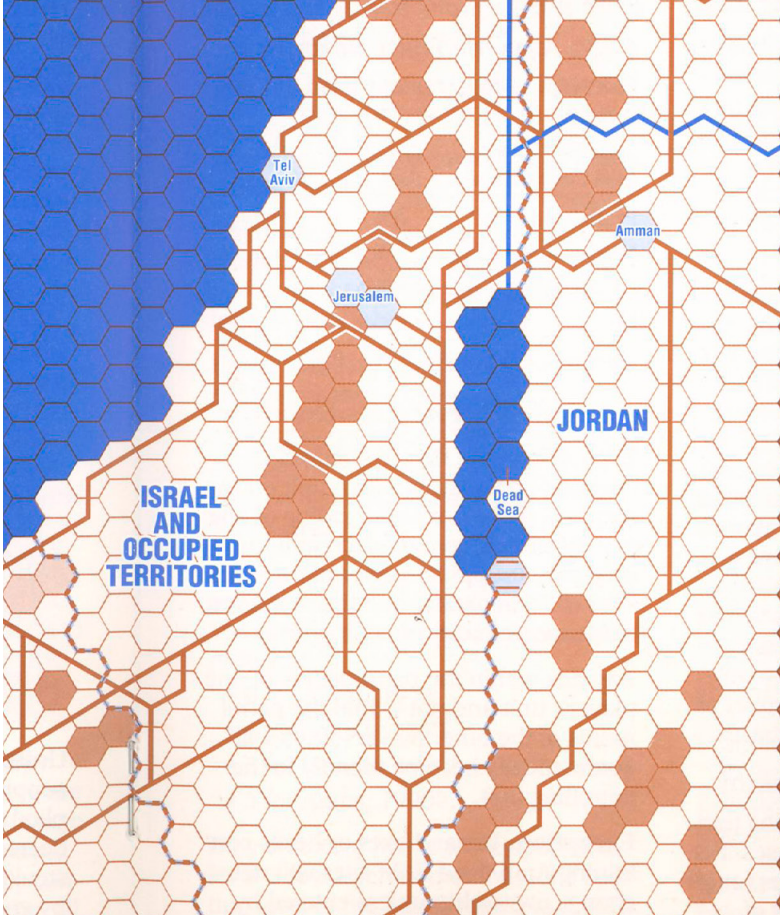
Haluamani pelin ainekset siis ovat jo tavallaan olemassa. *The Long Driven* karu realismi yhdistettynä *American Truck Simulatorin* liikennemekanismeihin ja *Flight Simulatorin* karttoihin olisi jo aika lähellä sitä, mitä reilut 35 vuotta sitten pihalla visioimme. Vielä kun joku tekisi sen.

JOKin PELIPALSTA

Pelihyllyn SSI-NURKAN herkkupaloja

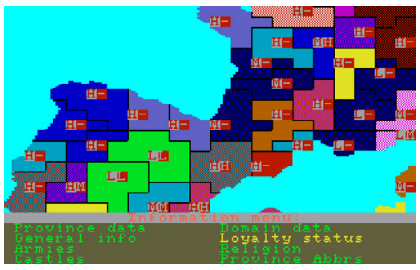
Joskus muinoin mietin onko mitään kauniimpaa kuin panssaridivisioona heksalla? No tietenkin kaksi panssaridivisioonaa!

Teksti: Jukka O. Kauppinen



Aikoinaan nykyiseen toimistooni muuttaessani minulla oli viimein mahdollisuus tehdä jotain mukavaa vanhalle pelikokoelmalleni. Olin säilönyt vanhoja arvostelukappaleitani roskasäkeissä kellari- ja vinttivarastoissa vuosikymmenestä ja muutosta toiseen. Ei se kaikkein kaunein ja arvokkain säilytystapa, mutta ainakaan pelit eivät päätyneet kaatopaikalle. Niitä on joskus mukava katsella, toisinaan avata – ja joskus ne toimivat myös inspiraationa.

Tällä kertaa pelipalsta kertoo minulle rakkaista strategiapeleistä, jotka löytyivät pelihyllystöni alatasolta, vasemmasta SSI-nurkasta.



Medieval Lords: Soldier Kings of Europe

Martin Campion / SSI | 1991 | C64, PC

Ruotsalaisen Paradox Entertainmentin *Crusader Kings* -sarja nauttii hämmentävää kulttimainetta. Se on kasvanut ja kypsynyt digitaaliseksi maailmanhistoriaversioksi Kauniista ja rohkeista, joskaan tv-sarjan käänteet eivät yllä lähellekään tietokoneistetun historiastrategian uskomattomia ja silti lähes uskottavia käänteitä. Myös **Sid Meierin** *Civilization* nojaa samankaltaiseen maailmanhistorian uudelleen rakentamiseen, joskin valtavasti suoraviivaistetummassa ja viihteellisemmässä muodossa.

SSI:n vuonna 1991 julkaisema strategiapeli *Medieval Lords* oli kuitenkin se ensimmäinen peli, joka suistui historiankuvauksellaan hulluuden syövereihin. Erilaisia maailmanvalloitusstrategioita oli toki ollut aiemminkin, mutta ei tällaisia. Yhdysvaltalaisen historian professorin **Martin Campionin** (JOKin pelipalsta, Skrolli 2019.1) suunnittelema ohjelma ei ollut enää strategiapeli: se oli avoimen hiekkalaatikon simulaatio historiasta. Herra oli suunnitellut ohjelman alkujaan omille historiantunneilleen, joten siinä oli muun muassa mahdollisuus kymmenelle ihmispelaajalle ja kuudelle tekoälyvastustajalle.

Mitä *Lordsissa* siten tehtiin? No, mitä nyt hallitsijat puuhastelivatkaan Euroopan villeinä vuosina 1028-1530? Rosvosivat, sotivat, liittoutuivat, polttivat kerettiläisiä, törsäsivät liikaa rahaa ja ruoskivat veroista nurisevia ja kapinoivia kansalaisia. Pelaajan sukupolvesta toiseen elävä neuvonantaja näki hallitsijan toisensa perään nousevan, kasvavan ja kuolevan, katsoen kenties kauhulla näiden jälkeläisten katastrofaalista idiotismia.

Tavallaan *Lords* oli aikaansa edellä. Commodore 64:lle ja PC:lle tehty historiallinen simulaatio loi maailmasta avoimen hiekkalaatikon, joka tulkitsi historiallisia syy-seuraus-tapahtumaketjuja uskottavasti, luoden joka pelistä omanlaisensa hiekkalaatikon. Mikään helppo välipalapeli se ei ollut. *Civilizationin* jättimenestys nojaa helppouteen, *Crusader Kingsit* puolestaan uskomattoman syvään ihmissuhde-, genetiikka- ja hallintosimulaatioon – mutta joskus kannattaa arvostaa tätä yhdysvaltalaista herraa, joka oli aikoinaan historiasimulaatiogenren valtias.

Red Lightning

Norm Koger / SSI | 1989 | PC, Amiga, Atari ST

Maailman paras pelisuunnittelija on tietenkin **Gary Grigsby**, mutta hänen kantapäillään seurasi pitkään toinen SSI-legenda, **Norm Koger**. Hänen peliensä kanssa yhteinen sotataipaleemme alkoi tulevaisuudessa, kun nousin *Stellar Crusaden* avaruustaivaille val-

Conflict: Middle East

Conflict: Korea

Kampfgruppe

taamaan galaksia paremman huomisen nimissä. Kuumottavaksi meininki muuttui, kun tähdet vaihtuivat kolmannen maailmansodan runtelemiin Keski-Euroopan kukkuloihin ja metsiin.

Euroopan valtaamista on saanut kokeilla monessa pelissä – esimerkiksi PSS:n *Theatre Europe* oli aikoinaan todella suosittu kuvaelma idän ja lännen suursodasta. *Redissä* siihen tarjottiin uudenlaista näkökulmaa ja syvyyttä, jossa kaikki istui kohdalleen. Sen mittakaava tarjosi mahdollisuuden sekä taktikointiin että suuriin manööverei-hin, kun Varsovan liiton ja Naton joukot kohtasivat taistelukentällä. Pelaaja saattoi sekä näpertää detaljitasolla, että keskittyä yksiköiden komentamiseen. Ilmavoimien osuus ja kansakuntien poliittisen selkärangan vahvuus suursodan keskellä toteutettiin nekin taitavasti. Samantapainen monitasoisuus näkyi myöhemmin Kogerin suurimmaksi hitiksi kasvaneessa *The Operational Art of War* -pelisarjassa.

Jännittävintä oli tietenkin pukea päähänsä punatähtinen lakki ja lähteä vapauttamaan kapitalistista länttä. **Tom Clancynsä** lukenut pelaaja viipaloi Naton linjoja ja eteni länteen – kohti Englannin kanaalia ja Pariisia. Muistan yhä etäisesti sen hyvän mielen tunteen, kun punaiset panssarini vyöryivät Amsterdamiin.

Helpolla tuokaan saavutus ei tullut. *Red Lightningissa* monipuolinen ja -mutkainen pelimekaniikka näet yhdistyi kauheaan käyttöliittymään. Olen muutaman kerran koittanut pelata *Rediä* uudelleen, edes vähän lämmikkeeksi, mutta ei. Se ei ole nykyihmisen peli. Valikkoramppaus on raivostuttavaa ja yksinkertaistenkin asioiden tekeminen vaatii hirveästi kliksuttelua ja pinnaa. Silloin aikoinaan tämän kuitenkin kesti, sillä pelin sodankuvaus oli niin tuore ja erilainen. Enää ei kykene.

Conflict: Middle East ja Conflict: Korea

Norm Koger / SSI | 1991 ja 1992 | PC, Amiga

Koger jatkoi hyväksi havaitulla linjalla seuraavissakin peleissään. *Red Lightningin* pelimoottorista jalostui *Conflict,* jonka kaksi peliä olivat edeltäjäänsä monipuolisempia ja samalla pelaajaystävällisempiä kokemuksia. Eivät nekään mitään koko kansan suosikkeja olleet, mutta ainakin räpeltämistä oli vähennetty tuntuvasti.

Molemmat *Conflictit* olivat taitavasti rakennettuja. Israelin sodat arabinaapureita vastaan (historiallinen 1973, hypoteettinen 1990) ja Korean sota (1950–1951) tiivistivät kiihkeät yhteenotot järkevään mittakaavaan ja värikkääseen sotahistorialliseen pelaamiseen. Sotaa käytiin niin aavikoilla, vuoristossa kuin ilmassa, poliittista kulmaa unohtamatta. Esimerkiksi Israelille on äärimmäisen tärkeää pitää huolta sekä oman väestön kestämisestä, että riittävästä sotamenestyksestä niin, ettei Jordania liity kamppailuun avaten kolmatta rintamaa. Korean sota on puolestaan niin täynnä äärimmäisiä käänteitä, että sen kokeminen pelimuodossa on melkoista vuoristorataa, etenkin jos historiankirjansa lukenut osaa edes vähän värittää taistelukentän tapahtumia korviensa välissä.

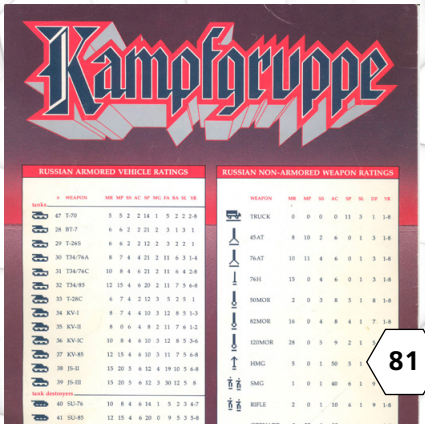
Conflictit ovat nykysilmin jo kohtalaisen pelattavia strategioita. Vähän kömpelöitä, mutta silti sutjakoita, etenkin jos käyttää vauhdikasta emulaattoria tai turbotettua Amigaa/Ataria. Vaikka en enää niille antaisikaan muinaista 95/100 pisteen arvosanaa, niin ovathan nämä aikansa laatustrategioita, joiden viehätys ei ole täysin kadonnut 31 vuodessa. Se on kova saavutus muinaispelille.

Kampfgruppe

Gary Grigsby / SSI | 1985/1988 | Amiga, C64, Apple II, Apple 8-bit, PC

Edellä mainitun Gary Grigsbyn tuotantoon mahtuu enemmän huikeita luomuksia kuin useimmilla nykypäivän pelifirmoilla on esittää koko repertuaarinaan. Yksi herran aikoinaan hienoimmista ja mullistavimmista peleistä oli historiallinen sotastrategia *Kampfgruppe*, joka toi peliruuduille ensimmäisen kerran taktisen mittakaavan sotaa sujuvassa, sutjakassa ja lautapelimäisessä muodossa. Grigsby oli luultavasti pelannut huolella Avalon Hillin *Squad Leaderiä*, sillä pienten joukkue- ja komppaniatasoisten osastojen komentaminen taistelukentällä oli hämmästyttävän toimivaa ja loogista.

Ja on edelleenkin. Pelin kasibitti- ja PC-versiot eivät ole käyttöliittymiltään parasta A-ryhmää, mutta vuoden 1988 Amiga-versio on aikalaisekseen todella hyvin suunniteltu sotapeli. Graafinen käyttöliittymä on selkeä ja helppokäyttöinen, eikä tärkeää informaatiota ole piilotettu. Pelilauta näyttää karulta, mutta on kiitettävän informatiivinen. Itärintaman suurtaistelujen sotiminen sujuu mukavasti ja nautinnollisesti yhä tänäänkin kokeiltuna, mikä on yllättävää näin iäkkäältä peliltä. Moni uudempi Amiga/Atari/PC-peli on kömpelömpi ja sekavampi pelattava, mutta Grigsbyn teos sen kun porskuttaa ja naureskelee mennessään Kogerin edellämainituille luomuksille. Kättä lippaan: Grigsby taitaa nämä jutut.



PELIEN HISTORIAA PODCAST-MUODOSSA

Podcastien kuuntelu on erinomainen tapa syventää retropeli-harrastustaan.

Teksti:
Eino Keskitalo

Pelaaja vanhenee ja nostalgia kalvaa, mutta ruuhkavuosina ei millään ehtisi perehtyä pelihistoriaan lukemalla tai edes YouTubea tuijottamalla. Puhumattakaan siitä, että ehtisi pelaamaan hyllyyn kertyviä aarteita! Onneksi työmatkat ja kotityöt ovat monia-jolle otollisia hetkiä, ja pelihistorian opintoja voi suorittaa kätevästi podcastien avulla. Ylen erinomaisia *Pelikonsolien* ja *Kotimikrojen kronikoita* ei tarvinne tässä enempää mainostaa, vaan voimme katsastella kansainvälistä tarjontaa...

Video Game History Hour

Nimensä puolesta ensimmäiseksi esiteltäväksi sopii *Video Game History Hour*. Podcastia julkaisee Video Game History Foundation, yhdysvaltalainen voittoa tavoittelematon säätiö, jonka tavoitteena on ”*säilyttää, juhlistaa ja opettaa videopelien historiaa*”. Säätiö julkaisee podcastia viikoittain, ja sen seuraaminen on viihdyttävä tapa pysyä kärryillä varsinkin yhdysvaltalaisen pelihistoriankirjoituksen tilasta.

Vieraana on usein sisällöntuottajia tai historioitsijoita, jotka ovat tehneet tutkimusta ja julkaisseet esimerkiksi Youtube-dokumentteja tai blogikirjoituksia eri aiheista. Aiheet voivat olla esimerkiksi pelejä (*Trespasser*), peliyrityksiä (DMA Design), laitteita (Nokia N-Gage) tai vaikka julkaisuja (*Nintendo Power*). Jaksoissa käsitellään itse aihetta, mutta myös aiheesta tehtyä tutkimusta. Pelintekijöiden haastattelut ovat vähemmistössä. Heistä vieraana ovat ehtineet käydä muun muassa **Garry Kitchen**, **Sid Meier** ja **Howard Scott Warshaw**.

Video Game History Hour käsittelee pelihistorian lisäksi paljon myös itse pelien ja pelihistorian säilyttämistä ja on tavallaan metapelihistoriapodcast. Ohjelman eräänä tavoitteena onkin hoksauttaa, että pelihistorioitsijaksi voi ryhtyä matalalla kynnyksellä. Podcastissa tehdään pelihistorian tuottamista näkyväksi ja innostetaan kuuntelijoita tarttumaan itse kutakin kiinnostaviin aiheisiin. Säätiö on keskittynyt varsinaisten pelien sijasta peleihin liittyvän materiaalin säilyttämiseen. Näistä esimerkkejä ovat lehdet, markkinointimateriaali ja muu paperitavara sekä lähdekoodi. Säätiö mainostaa toimivansa mielellään lähteenä matalalla kynnyksellä ”eritasoisille” tutkijoille – varmaan moni *Skrolliin* ja *Retro Rewindiinkin* pelijuttuja kirjoittava voisi hyvillä mielin olla säätiöön yhteydessä!

Poiminta: Jaksossa 39 haastateltavana on **Mat Van Rhoon** ja aiheena FMV-pelisarja *Tex Murphy*. Kiehtovia juttuja vanhan videomateriaalin etsimisestä ja arkistoinnista.

They Create Worlds

They Create Worlds on paitsi podcast myös blogi ja ennen kaikkea mittava, kaikenkattava kolmiosainen kirjasarja pelialan esihistoriasta 2000-luvulle asti. Kirjoista ensimmäinen on julkaistu vuonna 2019. Podcast ammentaa kirjoittaja **Alexander Smithin** kirjoja varten tekemästä tarkasta taustatyöstä sekä lukuisista haastatteluista, ja se on seikkaperäinen ja pikkutarkkakin. Onneksi Smith on myös hyvä tarinankertoja.

Erityisen kuuntelemisen arvoisen tästä podcastista tekee sen kokonaisvaltainen pelialan käsittely nimenomaan liiketoimintana. Usein pelihistoriassa keskitytään itse peleihin ja niiden kehittäjiin, ja yleensä liiketoiminnallisuutta käsitellään lähinnä ”ahneus rusentaa luovuuden” -näkökulmasta. Smith onkin keskittynyt omissa haastatteluissaan erityisesti liiketoimintapuolen ihmisiin, joiden tarinoita muut enemmän peleistä itsestään kiinnostuneet lähteet eivät ole samalla tavalla taltioineet. Missään nimessä pelintekijät tai tärkeiden pelien vaikutus pelien kehitykseen eivät jää huomiotta, mutta tarkastelu on tasapainoisempaa kuin keskimäärin. Olen kuuntelijana saanut They Create Worldsista tämän takia monipuolisemman käsityksen pelialan historiasta kuin olen osannut kaivatakaan.

Jaksot saattavat käsitellä yksittäisiä yrityksiä (pienistä suuriin), tiettyjä aikakausia pelialan eri puolilta (konsoli-, tietokone- tai kolikkopelit), pelisarjoja tai genrejä, tai vaikka keskittyä henkilöihin. Liikutaan siis eri tarkkuustasoilla, mutta aina on luvassa perusteellista tarinointia. Tekijöiden kunnia-asiana on välttää toistamasta vanhoja urbaanilegendoja, joten jaksot perustuvat aina tarkimpaan, ajantasaisimpaan mahdolliseen tietoon. Esimerkiksi tätä kirjoitettaessa vasta julkaistuista Nintendon esihistoria -jaksoista selviää, oliko Nintendo tosiaan mukana taksi-, love hotel- ja pikariisibisneksissä ennen videopelejä.

Poiminta: They Create Worldsissa on tarkasteltu Ataria useasti, pitkään ja monesta eri näkökulmasta. Jaksossa 32 on tästä kelpo esimerkki – Atari-brändin tarina kautta historian.

The Retro Hour

Siinä missä edellisten podcastien tekijät ovat selkeästi pelihistorijoitsijoita, on *The Retro Hour* retroharrastajien ylläpitämä podcast. Tyypillinen viikoittainen The Retro Hour -jakso sisältää tuoreimmat retrouutiset ja jonkun vanhan pelintekijän tai vaikkapa retroaiheisen sisällöntuottajan haastattelun. Se, että podcast on isobritannialainen, on piristävää, sillä muut tässä esitellyt podcastit ovat yhdysvaltalaisia.

Juontajien nörttijätkämäinen jutustelu on leppoisaa kuunneltavaa, ja vanhojen pelilegendojen haastatteluissa soi sopivasti fanihenkinen kunnioitus ja innostus. Yksi tekijöistä, **Dan Wood**, on myös radioalan ammattilainen, joten paketti on sopivan puoliammattimainen – ei amatöörimäinen, mutta rento. Retropelimaailman uutisia on hauska seurata podcastia kuuntelemalla, ja juontajat kommentoivat niitä viihdyttävästi.

The Retro Hour onkin etenkin kahta edellistä esiteltyä podcastia huomattavasti viihteellisempi, mutta tekee kuitenkin historiallisesti merkittävää työtä haastatellessaan viikoittain pelialan tekijöitä, joista moni on aloittanut työnsä jopa 70- tai viimeistään 80-luvulla. Esimerkiksi They Create Worldsin tekijät suosittelevat The Retro Hourin kuuntelemista. Pitää kunnioittaa juuri ennen retrobuumia aloittanutta podcastia, jolla on viikosta toiseen mielenkiintoinen vieras haastateltavanaan.

Poiminta: **Ian Grive** kertoo jaksossa 213 mehukkaasti Pysgnosis-yhtiön alkuvuosista.

Retronauts

Viihteellinen on myös *Retronauts*, joka on eräs retropelipodcastien arkkivaareista. Retronauts alkoi jo vuonna 2006 1UP-peliuutissivuston kylkiäisenä. Sivuston toiminnan lopettamisen jälkeen tekijät ovat jatkaneet podcastin tekemistä joukkorahoituksen voimin. Tekijöiden tausta ja ote tekemiseen ovat journalistisia. Siinä missä muut tässä esitellyt podcastit ovat tekijöilleen oheistoimintaa tai harrastus, Retronauts on selkeimmin tuote, joka on tekijöilleen ansiotyötä.

Retronautsin jaksot ovat yleensä paneelikeskusteluita. Keskustelijat ovat hyviä puhujia ja tuntevat aiheen. Aiheita ovat esimerkiksi yksittäiset pelit, pelisarjat, julkaisijat ja eri pelilaitteet. Esimerkiksi jaksot 2000-luvun alun pelikonsoleiden julkaisuista ovat oikein kiinnostavaa kuunneltavaa. Retronautit ovat tällöin olleet sopivassa iässä ja maalaavat jaksoissa kuvaa 9/11-aikakauden Yhdysvalloista. Pelejä muistellaan eskapismina tai pohditaan sitä, miten ne heijastelevat senaikaista maailmaa. Paneelikeskusteluiden lisäksi podcastissa kuullaan silloin tällöin myös hyviä haastatteluja. Joskus klassikkopelien tekijät pääsevät ääneen, mutta useammin haastatellaan ihmisiä hieman pienempien retro- tai retrohenkisten julkaisujen tiimoilta.

Retronauts on laadukas ja viihdyttävä osa Yhdysvaltalaista retropelipuheen kaanonia. Sillä on pitkä historia ja se on inspiroinut monia muitakin pelipodcasteja. Itse en juurikaan haikaile 1UP:n aikaisen Retronautsin tunnelmaan, sillä maailmassa on tarpeeksi nuorten kyynisten jätkien ironisen viileää nokkeluutta. Ohjelma on kypsynyt vanhetessaan, ja tekijäkaartia on laajennettu niin että esimerkiksi eurooppalaista näkökulmaa on saatu paremmin mukaan.

Poiminta: Alkuperäisen XBoxin julkaisua käsittelevä jakso (416) on hyvä esimerkki tyypillisestä Retronauts-paneelikeskustelusta, joka sitoo aiheen tapahtumahetken ajankuvaan.

Pixelated Audio

Mainitaan vielä pelimusiikkipodcast *Pixelated Audio*, joka keskittyy useimmissa jaksoissaan yksittäisen pelin ääniraitaan. Pelistä soitetaan valikoituja kappaleita ja juontajat kommentoivat niitä yleensä yleisellä tasolla, mutta välillä tehden kiinnostavia huomioita teknologisesta tai musiikin teoriaan liittyästä näkökulmasta. Vanhemmissa pelilaitteissa käytetyt äänipiirit saavat usein erikoishuomiota.

Pixelated Audio harvemmin käsittelee niitä tunnetuimpia *zeldoja* ja *mega maneja*, vaan sukeltaa ansiokkaasti syvään päätyyn. Miltä kuulostaa vaikkapa tanskalainen Colgaten mainospeli *Harald Hårdtand: Kampen om de rene tænder* tai eteläkorealainen DOS-peli *Illusion Blaze*? Entä Nintendo DS:n *1500DS Spirits: Igo*? Kuuntelemalla Pixelated Audiota se selviää.

Tälle listalle tämä podcast pääsee ennen kaikkea siksi, että ohjelmassa kuullaan usein alkuperäisten säveltäjien haastatteluja. Yhdellä juontajista, **Bryan Mosleylla**, on ässänä hihassaan hyvä japanin kielen taito. Haastateltavana onkin kuultu kiehtova setti japanilaisia säveltäjiä **Saori Kobayashista** (*Panzer Dragoon Saga*) **Fumihito Kasataniin** (*Alisia Dragoon*). Myös länsimaalaisia säveltäjiä vierailee show'ssa usein. Uudempien indie-äänimaisemien parista esimerkiksi voi nostaa **Megan McDuffeen** (*River City Girls*, klassikkojen puolelta mainittakoon **Chris Hüelsbeck**. Pixelated Audion eetoksesta kertoo jotain se, että kaikista Hüelsbeckin säveltämistä peleistä aiheena oli esimerkiksi *Turrican*-sarjan sijaan oudompi *Jim Power in Mutant Planet*.

Poiminta: Sympaattisessa jaksossa 103 ovat vieraina retroseikkailupeli *Dropsy: A Point and Click Hugventuren* kehittäjä **Jay Tholen** (Hypnospace Outlaw) ja säveltäjä **Chris Schlarb** (NightSky).

Linkit myös: skrolli.fi/numerot

Copyleft (GPL) Twitterissä

Kiitos kommentista. Käyttäjänoikeus on Skrolli 2021.4:n kolumnissani mainitun tekijänvelvollisuuden tapainen suomennos. Molemmat ovat alkuperäistä englanninkielistä copyleft-vitsiä parempia, sillä ne sentään kertovat oikean suuntaisesti tekijänoikeuksien painopisteen muutoksesta, mitä sana copyleft ei tee. Molemmat käännökset ovat silti sanaleikkejä sanasta tekijänoikeus, eivätkä viesti niitä erityisiä oikeuksia ja velvollisuuksia vapaista ohjelmistoista, joita copyleft-merkitykseen nykyisin liitetään.

Sana käyttäjänoikeus on myös vaarassa sekaantua käyttöoikeuteen ja käyttöoikeussopimuksiin, joissa oikeudet voivat vaihdella suuresti. Nähdäkseni paras termi sekä suomeksi että englanniksi olisi (ollut) sellainen, joka vapautuisi kokonaan sanaleikkien kahleista ja keskittyisi sanoittamaan merkitystään. GPL-manifestin "remain free" sanapari tuntuisi itsestäni parhaalta lähtökohdalta – toisin sanoen ohjelmiston vapauden pysyvyys.

Käännöksen ymmärrettävyyttä voi kokeilla sijoittamalla se GPL:n (*GNU General Public License*) aloituslauseeseen: GPL "is a free, copyleft license for software..."

GPL on vapaa käyttäjänoikeuslisenssi ohjelmistoille? vai GPL on lisenssi pysyvästi vapaille ohjelmistoille?

Janne Sirén

Taksonomiaa

Unohtaisin copyleftin suoraan suomentamisen, ja kehittäisin seuraavan taksonomian: muokkausvapaa ja julkaisuvapaa. Ajatuksena se, että esim GPL2 on muokkausvapaa, mutta ei julkaisuvapaa lisenssi. BSD taas on muokkaus- ja julkaisuvapaa. Ehkä näitä "taksoneita" voi vielä hioa, mut mielipiteeni on että [edellä mainitun] artikkelin ehdotuksista lähdettiin katsomaan suomentamista liian suoraviivaisesta kulmasta.

jarkko, #skrolli (IRCnet)

Itsekin taivun ajattelemaan, että sanan kääntämisen vaikeus tässä paljasti, kuinka asiaa on syytä lähestyä ihan toisesta kulmasta. Hyviä kulmia on varmasti useampia. Toki ajattelen lisäksi, että alkuperäinen englanninkielinen sana copyleft on tässä se suurin synti...

Janne Sirén

Oikaisuja ja huomioita

Skrolli 2021.4:n paperiversiossa Skrolli 6,4 vuotta sitten -palstalla (s. 46) oli virhe painettujen sivujen määrässä. Virhettä ei ole pdf-lehdessä. Skrollin painettujen sivujen määrä ylitti tuolloin 9,5 miljoonaa sivua, ja 2022.1:n myötä määrä nousee noin 10 miljoonaan. Luvuissa eivät ole mukana tulostettu Skrolli 2012.0 -näytenumero, puhtaasti digijulkaistu Skrolli 2016.1E Sample tai muut digitaaliset lehdet.

Skrolli 2021.3:n Postipalstalla (s. 102) oli sitaatti, jonka jälkipuolisko kuvaili kuuden vapausasteen käsitettä VR-tuoleissa. Lehdestä puuttui kuitenkin tekstissä mainittu, kuvauksen osoittanut kursivointi: "Raskaimmat mallit liikkuvat peräti kuudessa vapausasteessa: eteen, taakse, sivuille, ylös ja alas. *Ajamisen matkimisen lisäksi niiden luvataan kallistuvan, kääntyvän ja nyökkäävän kuin lentokoneet sekä kohoavan, huojuvan ja työntyvän kuin laivat.*"

Janne Sirén

Tietoyrityksiä

Vielä 1970-luvun alussa tietokoneen hankinta oli useille keskisuurille yrityksille liian kallista, joten yritykset päätyivät yhdessä toisten paikallisten yritysten kanssa perustamaan erillisiä yrityksiä tuottamaan tietotekniikan palveluita. 1970-luvulla alussa perustettiin useita Tieto-alkuisia yrityksiä: Tietotehdas, Tietonauha, Tietosavo, Tietojyvä, Tietobotnia...

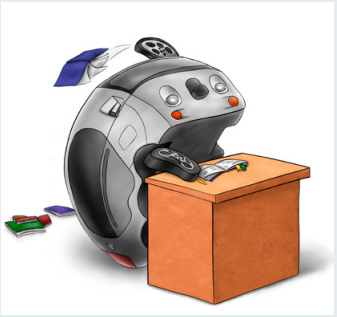
Yritysten tehtävänä oli aluksi palvella omistajayrityksiä tietotekniikan palveluissa ja myöhemmin ne laajensivat asiakkuuksiaan myös omistajien ulkopuolisiin yrityksiin. **Pekka Salojärvi** kirjoitti Tietojyvän perustamisesta Keskisuomalainen-sanomalehdessä 22.1.2007:

"Jyväskylän merkittävimmät yritykset mm. Are, Gummerrus, Sohlberg ja Keskisuomalainen päättivät suostua IBM:n suostutteluun ja perustaa Tietojyvän (1973). Toimitusjohtajaksi valittiin **Tauno Rantanen** ja minä sain kunnian ryhtyä hallituksen puheenjohtajaksi. Osakkaiden lähtökohtana oli saada ostettua tietokonepalvelut Jyväskylästä. Omiin laitteisiin ei yksittäisellä yrityksellä ollut varaa."

"Toiminta siirtyi Gummeruksen tiloihin vuosikymmenen lopussa ja parhaimmillaan se työllisti yli 100 henkilöä. Tietojyvä myytiin Tiedolle 1984."

Mauno Poikonen

Mauno on jyväskyläläisen Suomen tietokonemuseon (suomentietokonemuseo.fi) puheenjohtaja. Museo teki hiljattain haastattelun kolmen Tietojyvän työntekijän kanssa. He kertoivat millaista oli tehdä töitä 1970-luvun eräajokoneella – haastattelun ja muun virtuaalisestikin saavutettavan museonäyttelyn linkit Skrollin verkkojatkoilla (skrolli.fi/numerot). Maunon historia-anekdootteja voi ja kannattaa muutenkin seurata Facebookin Suomalaisen tietotekniikan historiaa -ryhmässä (facebook.com/groups/733326956836475).



Kuva: Susanna Viljanmaa

Päivitys: R-kieli – Avointa ajokilometridataa

Käsittelin Skrollin numerossa 2021.3 Traficomin avointa ajoneuvodataa R-ohjelmointikielellä. Artikkelista selvisi muun muassa, että vain muutamalla prosentilla liikennekäytössä olevista henkilöautoista on ajettu yli puoli miljoonaa kilometriä. Spekuloin kuvioiden ja keskiarvojen perusteella, että keskimääräinen henkilöauton elinikä kilometreinä olisi vajaa 300 000 km. Tavoitin jutun jälkeen Traficomin erityisasiantuntija **Pasi Kilpelän**, joka pääsi käsiksi romutettujen autojen tilastotietoihin. Niiden mukaan tammi-heinäkuussa 2021 romutettujen henkilöautojen keskimääräinen mittarinlukema oli 300 772 kilometriä. Romutettaessa ei kirjata kilometrilukemaa muistiin, eli kilometrimäärä on edellisestä katsastuksesta.

Kilpelän mukaan romutettujen autojen data on haasteellista ja vaatii käsittelyä: romutettavat autot ovat usein olleet pitkään poissa liikennekäytössä ilman katsastusta ja viimeisin katsastus voi olla jopa ennen kuin mittarilukemia alettiin tallentamaan ajoneuvorekisteriin. Lisäksi monilla vanhoilla henkilöautoilla ei ole niin kutsuttua miljoonamittaria eli ei tiedetä, miten monta kertaa mittari on pyörähtänyt ympäri ja/tai onko sitä "rullattu". Osalla autoista on myös mailimittarit.

Mukana keskiarvossa ovat autot, joilla ajosuorite on ollut keskimäärin yli 7 000 km vuodessa käyttöönottovuodesta lähtien. Valinnan teki Tilastokeskus dataa tutkiessaan ja puhdistaessaan – todennäköisesti syynä on se, että näin saatiin siivottua pois autot, jotka eivät ole olleet käytössä ollenkaan tai joiden lukemia ei ole kirjattu.

Vesa Linja-aho

Lisää Suomesta: Moonbound (3DS)

Skrollissa 2021.3 oli artikkeli maailmalle viime vuosina päässeestä niche-Suomi-pelistä, *Smarty and the Nasty Gluttonsista* (Amiga). Vaikka Nintendo 3DS ei olekaan aivan Amigan iässä, alun perin vuonna 2011 julkaistu käsikonsoli on sekin nykymittapuulla jo melko iäkäs tapaus. Niinpä 3DS-pelien digitaalijakelukin alkaa olla ehtoopuolella. Jakeluun ilmestyy enää lähinnä karkeatekoisen oloisia pikkupelejä. Vinkkinä kuitenkin Skrollin lukijoille: yksi näistä on kiinnostava kotimainen tapaus.

Moonbound (moonbound-game.com) on **Jouni Mutasen** kehittämä peli Nintendo 3DS:lle. Se julkaistiin Nintendon eShopissa Yhdysvaltojen puolella vuoden 2021 lopussa ja Euroopassa peli tuli saataville tammikuussa 2022. Ensivilkaisulla Moonbound ei erotu 3DS:n pikkupelitarjonnasta – ruutukaappauksissa grafiikka on hieman valjua. Kotimaisuus sekä Discordissa nähty kehaisu riittivät kuitenkin ylittämään oman ostokynnykseni.

Ilokseni onkin todettava, että peli tuntuu käsissä erittäin hyvältä. Mieleeni tulee ZX Spectrumin Jet Pac ylävinkkelistä tarkasteltuna. Toisaalta peli on kuin Pac-Man-vaikutteinen kolikkopeli 1980-luvun alusta. Moonboundissa ohjastetaan avaruuspukuista hahmoa yhden ruudullisen kokoisissa, yllhäältä kuvatuissa kentissä. Pelaaja kerää simahtaneeseen avaruusalukseen polttoainekristalleja ja välttelee hirviöitä. Avaruuspuvun happivarasto toimii myös rakettirepun polttoaineena, ja sitä pitääkin osata palata taktisesti täydentämään aluksen luokse.

Kentissä liikkuu erilaisia möllejä, jotka käyvät nopeasti päälle saatuaan pelaajasta vihiä. Maastossa on seiniä, happolammikoita, piikkikasveja ynnä muuta. Näihin ei saa astua, mutta toisaalta seinä on seinä useimmille vastustajillekin. Vastustajat käyttäytyvät ja hyökkäävät eri tavoin, joten kohtaamisissa pitää käyttää nopeaa harkintaa. Rakettirepulla voi oikoa mutkat ja välttää vaarat, mutta sen käyttö on rajoitettua.

Pelattavuuden elementit ovat hyvin tasapainossa, ja kentän yläpuolella lentäminen on hieno lisä genreen, jossa voisi kuvitella kokeillun jo kaikkea. Peliin pääsee sisään nopeasti, kunhan saa rytmistä kiinni: tarkkaile missä hirviöt ovat, suhauta seuraavan kristallin luokse rakettirepulla ja palaa alukselle nopeasti ennen kuin happi loppuu. Tarvittaessa jalkaudu taktisesti. Se säästää happea, mutta tekee pelaajasta myös haavoittuvan.

Puolustautumisvälineeksi on tarjolla vain tainnutuspyssy. Sen osuma jähmettää useimmat vihollisista hyväksi aikaa paikoilleen, muttei missään nimessä tee niistä vaarattomia. Kaikki tämä on ihanan stressaavaa, ja Moonbound on juuri oikealla tavalla turhauttava peli. Käytössä on yhdeksän elämää ja loputtomat jatkoyritykset – molempia sain helposti kulumaan runsain määrin, mutta uudestaan yrittäminen on vaivatonta.

Pelissä on kevyt taustatarina, ja vaikka tällaisia pelejä harvoin pelataan tarinan takia, sen läpinaksuttelu kehystää ja rytmittää toimintaa mukavasti. Myös **Juha Salinmäen** musiikki täyttää tehtävänsä erinomaisesti.

Eino Keskitalo

Kuva: moonbound-game.com

Tiesitkö, että... Skrolli 2021.4:n kannessa on tyylitelmä Amiga 500 Plus (1991) -kotitietokoneen emolevyltä. Keltainen suorakaide on Paula-äänipiiri ja turkoosi neliö "Super Fat" Agnus.

Palautekyselyn tuloksia

Skrolli järjesti verkon puolella Google Forms -palautekyselyn Skrolli 2021.4:n suosikkijutuista sekä lehden kannen ja sisällön kouluarvosanoista. Tässä tulokset tammikuun loppuun mennessä. Suosituimmat (kolmannes tai yli äänesti):

Jyrki J. J. Kasvi 1964-2021 (60 %)
Kolumni: Janne Sirén (48 %)
Huonoin kotitietokone (48 %)
JOKstoriaa: lehdenteosta ja... (48 %)
Commodore/Spectrum-sisäpiiri (44 %)
Turrikaanien yö (44 %)
Protokyberpunkin jäljillä (36 %)
Janne Sirén – Taka-ajatus (36 %)
Postipalsta (36 %)
Applen historian käännepisteitä (36 %)

Kannen / sisällön arvosana: 8,00 / 8,24 (n = 25).

Seuraavassa lisäksi tiivistelmiä palautekyselyvastauksissa ja muissa yhteydenotoissa lähetetystä kirjallisesta palautteesta:

"Joku täysin turha mutta teknologialtaan eksoottinen rakenteluohje olisi kiva. Tarvitsisin tekosyyn kolvin kuumentamiseen pitkästä aikaa."

"Sopivaa huumoria siellä täällä lämmittää mukavasti. Kiitos taas yhdestä numerosta, joka tuli luettua kokonaan!"

"Amiga-emolevyjä koskeva artikkeli oli paras."

"Skrollit tulee luettua aika tarkkaan. Luen taas hitaasti nautiskellen. Yleinen taso jutuissa on tosi hyvä! Tykkään siitä Sirénin palstasta. Kiitos koko Skrollin porukalle hienosta lehdestä!"

"Manun kummat vekottimet saisi korkeimman arvosanan minulta, jos [Retro Rewind -liite] olisi mukana äänestyksessä."

"Skrollissa 2021.2 kerrottiin numerossa 2014.4 esitellystä 12-vuotiaasta RasPi-ohjelmoijasta. Herran nimikirjaimet ovat A. L. Mitähän mahtaa kaverille nykyään kuulua ja onko hänen ohjelmoimat pelit jossain jakelussa? Uusintahaastattelu olisi mielestäni erittäinkin aiheellinen."

Suuri kiitos palautteesta. Lisäpalaute on tervetullutta osoitteessa skrolli.fi/palaute sekä Skrollin lukijakanavien kautta, jotka löytyvät osoitteesta skrolli.fi/pikaviestimet (Discord, IRC, Matrix). Ohessa myös suora QR-koodi ja kutsulinkki Skrollin Discordiin, jossa on yli 500 jäsentä.[1] Tervetuloa!

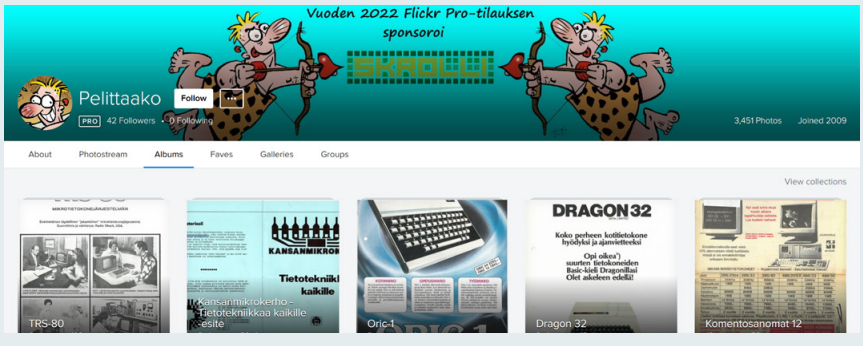
Janne Sirén

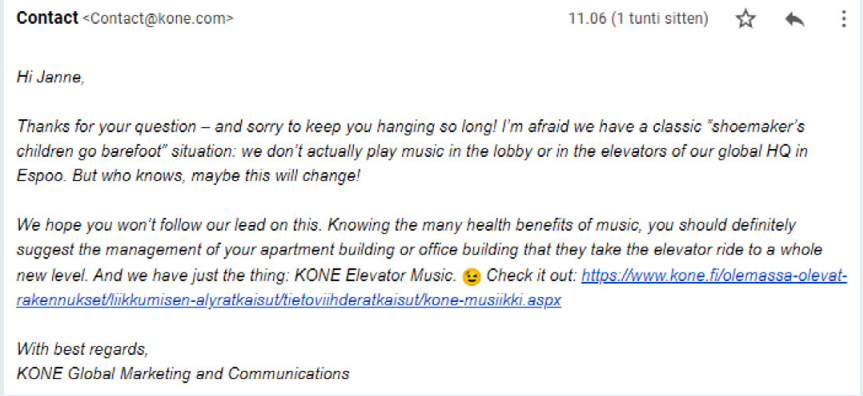
[1] Skrollin tuoreustakuu: Jäsenmäärässä vain 2.3.2022 ja edeltäneen viikon aikana Skrollin Discordissa käyneet tilit. Epäaktiiviset on karsittu. discord.com/invite/zG3P6VrHjW

Pelittaako – ATK-paperit digitaalisiksi

Janne Rautiainen on Suomen mikrotietokonehistorian paperijäljen pelastaja. Janne on erikoistunut haalimaan yrityksiltä, konkareilta ja keräilijöiltä historiallisia ATK-dokumentteja skannattavaksi, jotka hän sitten taltioi sähköisesti ja jakaa *Pelittaako*-arkistossaan. "Pätkis" ja kuvasitaatit kokoelmasta ovat tuttu näky myös Skrollin lukijakanavilla (skrolli.fi/pikaviestimet).

Skrollille lähettämässään kuvassa (vas. alla) hän on tosin harhautunut selvittämään mitä hissimusiikkia hissiyrityksen hisseissä soi, mutta lupaamme, että varsinaisessa dokumenttikokoelmassa on historiallisia tietokone- ja pelikuvastoja, jälleenmyyjien kuten Komentokeskuksen ja Infon uutiskirjeitä sekä muita tietoteknisiä papereita. Janne on myös ikuistanut kaikki Printtilehdet verkkoon julkaisijan luvalla. Skrolli sponsoroi Pelittaako-arkistoa taas tänäkin vuonna: flickr.com/photos/pelittaako/





Contact <Contact@kone.com> 11.06 (1 tunti sitten)

Hi Janne,

Thanks for your question – and sorry to keep you hanging so long! I'm afraid we have a classic "shoemaker's children go barefoot" situation: we don't actually play music in the lobby or in the elevators of our global HQ in Espoo. But who knows, maybe this will change!

We hope you won't follow our lead on this. Knowing the many health benefits of music, you should definitely suggest the management of your apartment building or office building that they take the elevator ride to a whole new level. And we have just the thing: KONE Elevator Music. 😉 Check it out: https://www.kone.fi/olemassa-olevat-rakennukset/liikkumisen-alyratkaisut/tietoviihderatkaisut/kone-musiikki.aspx

With best regards,
KONE Global Marketing and Communications

Kuva: Janne Rautiainen

Kuva: Janne Sirén

Aina yhtä avulias Skrolli-trolli kehotti hissifirmaa ottamaan mallia Tukholman Clarion Sign -hotellin SingStar-hissistä, josta oheistamme kuvan. Lisäpisteitä saisi kuulemma Out Run -hissistä, jossa on ratti kappaleen vaihtoon.

Podcast-painos

Olisiko mahdollista saada Skrollin jutuista myös podcast-versioita, eli jonkun lukemana pidemmät artikkelit. Olisi mukava kuunnella esimerkiksi ajaessa ja toisaalta moni vanha juttu saisi uuden elämän podcastina.

Saminyk

Sultant of Non

Sultant of Con ja *Huonoin kotitietokone* [Skrolli 2021.4] oli kirjoitettu jotenkin tarpeettoman negatiiviseen sävyyn. Lieköö yritys hauskuuttaa lukijaa vai mistä lie kysymys, mutta kummassakin ladeltiin lähinnä itsestäänselvyyksiä vailla varsinaista sisältöä tai lisäarvoa, ja vieläpä varsin katkeraan ja sarkastiseen sävyyn. Ei jatkoon.

Poiminta palautekyselyn vastauksista

Sultant of Conin osalta todettakoon, että kun on tarpeeksi marinoitunut IT-elämässä ja nähnyt työn todellisuuden, konsulttikaan ei aina jaksa silotella totuutta. Vaihtoehtona on, että höpisisin jossain lajitovereideni kanssa soijalattesta ja siitä, kuinka devaajat arvostavat enemmän merkityksellisiä pro bono -hommia kuin palkkaa, jolla tulee toimeen Itä-Suomen asuinkeskittymässä… Itse näen tähdellisemmäksi totuuden toitottamisen.

Mutta palautetta on aina hieno saada! En tosin ennakoi muuttavani kolumnieni sävyä olennaisesti. Jos se ei miellytä, en pahastu, jos kirjoitelmieni yli hypätään tai ne askarrellaan saksilla pois pahaa mieltä levittämästä. Sen sijaan jopa ilahtuisin palautepostista, jossa on konkreettisia vastaehdotuksia Suomen IT-alan kehittämiseksi.

Sultant of Con

P.S. Huonoin kotitietokone on tietenkin MSX, tuo Sultantin kotikylän konesotien ikuinen häviäjä. Tällä faktalla ei ole mitään tekemistä sen kanssa, että kylällä liikkui Salamanderista vain MSX-versio – en minä sitä olisi halunnutkaan pelata. C64 sen sijaan on melkein yhtä hyvä kuin se ainoa oikea Amiga eli OCS-Amiga.

Kuin kaksi konsolia

N-Gage ja Xbox vuosimallia 2021 (Espoo, Suomi / Waterloo, Belgia).

Kuvat: Janne Sirén

Komponenttipula? Syököötkakkua!

Skrollin sivumäärä

Mikähän on Skrollin sivumäärän teoreettinen yläraja… Onko tähän asti aina painettu kaikki materiaali mitä kuhunkin numeroon on saatu?

Eino Keskitalo

Lähes aina. Talkoohenkisenä lehtenä Skrolli on ennen kaikkea kokoelma siitä materiaalista, joka kuhunkin numeroon saadaan. Teknisten tai toimituksellisten syiden vaikutus sivumäärään on vähäinen ja rajoittuu enimmäkseen taittotekniikkaan ja yksittäistapauksiin, joissa jokin artikkeli siirretään. Kuten Skrollin 2021.4:n pääkirjoituksessa mainitaan, teoreettinen sivumäärämaksimi saattaa olla 148 sivua, koska tällainen lehti mahtuu 500 gramman postikuoreen. Otamme kuitenkin mielellämme vastaan rajaa haastavia artikkelivyöryjä osoitteessa toimitus@skrolli.fi.

Janne Sirén
(tiivistelmä Skrollin #lehdentekolukijakanavan keskustelusta)

Palautekyselyitä sähköpostiin?

Pari numeroa sitten linkki lukijapalautelomakkeeseen lähetettiin sähköpostiin. Luulen, että vastausprosentit romahtivat, kun palaute siirtyi vain Discordiin. Itselläni on teknisiä rajoitteita, joiden takia Discordiin pääsy on hankalaa, enkä sitä huvikseni käytä. Olisi mukava, jos palautelinkki löytyisi jatkossa myös vaikka Skrollin websivulta tai muualta vähemmän vaivan takaa.

Poiminta palautekyselyn vastauksista

Kiitos kommentista. Skrolli-palautteen kyseleminen vaihtelee numerosta toiseen ja välillä sitä ei tehdä lainkaan, mutta emme ole missään vaiheessa siirtyneet vain Discordiin. Palautelinkkimme löytyvät sekä osoitteesta skrolli.fi/palaute että sosiaalisen median kanaviltamme, joskin somejättien algoritmit kampittavat niiden jakelua ja ne voivat jäädä osalta näkemättä.

Viimeisestä kahdesta Skrollista tehdyn palautekyselyn linkki julkaistiin skrolli.fi:n lisäksi Facebookissa ja Twitterissä, kahden web-foorumin Skrolli-ketjuissa sekä pikaviestimissä (Discord, IRC ja Matrix). Tämä on ollut viime vuosina varsin tyypillinen jakelu. Vuosi sitten kokeilimme lisäksi lähettää kolmen numeron palautelinkit sähköpostitse. Kokeilu onnistui, mutta emme ole käyttäneet sähköpostia jatkuvasti, ettemme lipsuisi roskapostitteluun. Sähköpostiviestinnän kehittämistä kyllä mietitään.

Suosiotaan kasvattavat pikaviestimet, kuten Discord, vaikuttavat kannaltamme lupaavilta. Ne ovat kyselylomakkeita kaksisuuntaisempia ja toimivat luontevammin mobiilikäytössä, joka on nykyään yleistä. Kanavat mahdollistavat sekä lukijoiden kuulemisen että tekijöiden etsimisen. Perinteiseen sosiaaliseen mediaan verrattuna algoritmit myös rajoittavat niitä (toistaiseksi) vä-

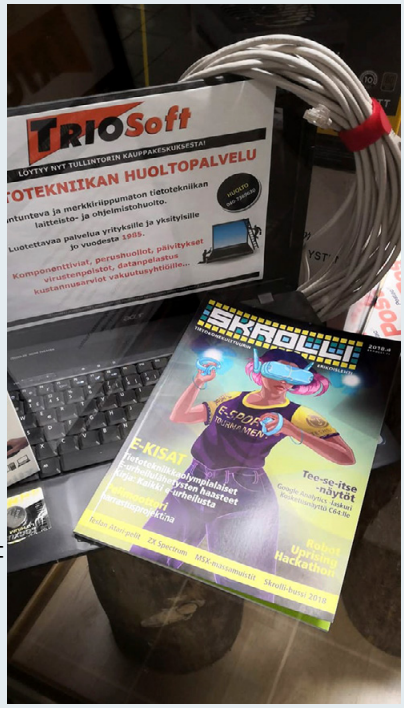
Kuva: Janne Sirén

Skrollia saa muualtakin kuin kioskeista ja marketeista

Pro-vinkki: Skrollit löytää yleensä hyllypaikasta, josta lehtiä on huvennut aukoksi asti. ;) Yläkuvassa Skrolli 2021.4 legendaarisessa Akateemisessa kirjakaupassa, Helsingin keskustassa, Alvar Aallon suunnittelemassa Kirjatalossa.

Kuva: Janne Sirén

Tyylikkäästi, kiireettömästi: Skrolli 2021.3 #megaturbo Suomalaisen kirjakaupan kahvilassa, osoitteessa Aleksanterinkatu 15 – myöskin Helsingin keskustassa.



Kuva: Jukka O. Kauppinen

Skrolli 2018.4 näyteikkunassa: Kaikkein legendaarisin ja pitkäaikaisin myyntipisteemme on Tampereen Triosoft. Kyllä, juuri se Triosoft. Myy edelleen mikrot ja Skrollit.

hemmän. Jos Discord ei ole oma juttu, kanaville pääsee lisäksi avointen Matrix- ja IRC-siltojen kautta (skrolli.fi/pikaviestimet). Etenkin Discordia ja Matrixia voi käyttää helposti selaimellakin.

Käyttötottumukset elävät ja yritämme olla siellä missä ihmisetkin. Uudet ihmiset ovat tällaisten talkooprojektien elinehto. Meidät tavoittaa kuitenkin jatkossakin myös muualta kuin pikaviestimistä. Välineet riippuvat lähinnä tekijöistä. Otamme mielellämme vastaan lisää yhteydenpitoideoita sekä niiden toteuttajia osoitteessa toimitus@skrolli.fi.

Janne Sirén

Digilehden loppuraportti

Skrollin mobiilisovelluksen käyttö päättyi viiden vuoden jälkeen joulukuussa, kun siirryimme digissä puhtaaseen pdf-jakeluun (ks. Skrolli 2021.2, s. 81). Kysyntä sovelluksessa luettavalle lehdelle oli pientä, kuten lopulliset myyntiluvut 2017–2021 kertovat. 2020-vuosikerrat: **25 kpl**, 2019: **31**, 2018: **24**, 2017: **42**, 2013–2016-paketit: **67**, suomi-irtonumerot: **159**, englanninkieliset: **54**. Androidin osuus näistä oli 143, iOS:n 259.

Skrollin toimitus

Kvartaalin kirje

Kvartaalin kirje tulee tällä kertaa kuvallisessa muodossa MikroBitti-lehden pitkäaikaisen toimittajan (1986–2004) ja päätoimittajan (1994–2004) **Markku Alasen** perikunnalta. Ensimmäisessä kuvassa on MikroBitti- ja Tietokone-lehtien työpaikkailmoitus vuoden 1986 alusta. Huomatkaa Markun tekemä alleviivaus toimittajan kompetenssivaatimuksesta, jota voi nykymittapuulla pitää poikkeuksellisen hienona. Tämän ilmoitukseen myötä vuonna 1985 Helsingin yliopistolta molekyyligenetiikan tohtoriksi väitellystä Markusta tulikin isona tutkijan sijaan toimittaja. Loppu on historiaa.

Toisessa kuvassa on Markku päätoimittajataipaleensa alussa sekä toinen aivan olennainen henkilö MikroBitin menestystarinan takaa: työpaikkailmoituksessakin mainittu toimitussihteeri **Riitta Tumppila**. Markun muistiinpanojen mukaan kuva on otettu 28.11.1994 Cantina West -ravintolassa, Helsingissä, mitä ei ole vaikea uskoa.

Muukin toimitus on mukana – on tärkeä hetki. Huomaa Markun uunituore MBnet-paita. Itseasiassa Riitallakin on sellainen päällä ja lähes kaikilla ▶

Kuvat: Janne Sirén

Pelit-lehti 30 vuotta

Skrolli toivottaa paljon onnea!

◀ Niko Nirvi on kertonut, että kannen "liian rajut" epäkuolleet peitettiin kustantajan johdon käskystä.

Tiesitkö, että...

Pelit-lehti syntyi C=lehden (1987–1/1992) tilalle. Sama tiimi teki näin vuodelle 1992 kaksi ykkösnumeroa. Pelit-lehti sai toisen puolen perinnöstään MikroBitin ja C=lehden Pelit-vuosikirjoilta (1987–1991), jotka se myös korvasi. C=lehdet (ja Pelit-vuosikirja) olivat muuten A4-kokoisia, kuten Skrolli, mikä on aikakauslehdelle edelleen harvinaista. Useimmat lehdet, kuten Pelit, ovat matalampia tai leveämpiä.

Exclusive Skrolli VIP Club

Jos jotain [Skrollissa] toivoisin, niin web-foorumia (tai vaikka Discordin osiota), johon eivät pääse kuin verifioidut tilaajat. Exclusive Skrolli VIP Club.

Shrike, #skrolli (IRCnet)

UUDEN AJAN ERIKOISLEHTITALO

TECNOPRESS OY

TECNOPRESS OY kustantaa hifi- ja videoalan erikoislehteä HIFIÄ, kaikkien kotimikron käyttäjien erikoislehteä MikroBITTIÄ, kaikille tietokoneen tarvitsijoille suunnattua ammattilehteä TIETOKONETTA sekä elektroniikan ja automaation ammattilehteä PROSESSORIA. Kaikki nolja lehteä edustavat alojensa huippua.
TECNOPRESS OY on SANOMA OSAKEYHTIÖN omistama uuden ajan erikoislehtitalo, joka työllistää Pitäjänmäellä sijaitsevissa tiloissa 30 henkilöä. Budjetoitu liikevaihtomme vuodelle 1986 on 17,5 milj. markkaa.

Toimittaja

MikroBITTI

Jos tunnet alan omaksesi, olet harrastanut kotitietokoneita jo pidempään, tunnet kenties useammankin kuin yhden mikromerkin paremmin kuin ulkonäöltä ja osaat ilmaista itseäsi kirjallisesti, tämä paikka on juuri sinulle.

Edellytämme hakijoilta hyvää kotitietokonealan tuntemusta, innokkuutta ja avointa suhtautumista kaikkeen uuteen sekä vähintään 18 vuoden ikää. Perehtyneisyys konekieliohjelmointiin on myös tarpeen.

Annamme tehtävään valitulle todella mielenkiintoisen toimen, jossa pysyy kehityksen kärjessä, oppii uusia asioita ja jossa voi perehtyä kotitietokonealaan pohjia myöten. Lisätietoja saat päätoimittaja Eskoensio Pipatilta, puh. (90) 120 5740.

Toimituksen sihteeri

TIETOKONE

Tietokone-lehti jäi ilman sihteeriä, kun Tumppilan Riitta siirtyi yhtiömme uusimman lehden, MikroBITIN, toimitussihteeriksi. Tarvitsemme työryhmäämme nyt uuden sihteerin. Tule sinä nuori HSO-sihteeri, jolla on jo jonkin verran työkokemusta, yhteistyökykyä ja olet kiinnostunut lehden tekemisestä. Tehtäviisi kuuluvat vaihtelevat sihteerintyöt, jotka suurelta osin liittyvät lehteen ja sen toimittamiseen.

Tarjoamme sinulle virkeän toimitusympäristön, mukavat työtoverit ja vakaan yrityksen henkilöstöedut. Kysy lisätietoja toimituspäällikkö Kaija Uimoselta, puh. (90) 120 5761.

Molemmat yllämainitut toimet ovat sekä miesten että naisten haettavissa. Hakemukset pyydämme lähettämään 1.4.1986 mennessä päätoimittaja Eskoensio Pipatille osoitteella Tecnopress Oy, PL 64, 00381 Helsinki. Kuoreen merkintä "toimittaja" tai "toimituksen sihteeri".

muillakin. Maailman suurimmaksi kasvava purkki MBnet avataan joulukuun lehdessä. Markkinointi & Mainonta tulisi aikanaan nimeämään MBnetin Suomen 9. arvostetuimmaksi verkkobrändiksi. Loppu on todellakin historiaa. Voit lukea Markun tarinan ja Markun omia sanoja Skrollista 2019.3.

Janne "MikroBitin Amiga-Janne" Sirén

Kuvat: Markku Alasen perhealbumi. Kiitos **Ninnu Koskenalho**/Markku Alasen perikunta. Kiitos myös Riitta Tumppila. Verkkojatkoiltamme (skrolli.fi/numerot) löydät Skrolli 2019.3:n pdf-lehden maksutta sekä linkkejä Skrollin Discord-lukijakanavien keskusteluihin, joissa on lisää nostoja MikroBitti-muistoista. Tarjolla on muun muassa **Wallun** spontaani piirrostulkinta ryyppäävästä Nalle Puhista, jonka hän kertoi piirtäneensä saatuaan potkut Nalle Puh -lehdestä.

osa XX: Suomen pelimuseon synttärien all-stars

Teksti ja kuvat: Jukka O. Kauppinen

Suomalainen pelijournalismi on asia, joka on vain tapahtunut. Sitä on tehty, mutta sitä on vasta aloitettu analysoimaan syvällisemmin. Pelimuseon viisivuotissyntymäpäivillä aihetta käsiteltiin sentään vähän lisää, alan sisältä käsin.

Koronan johdosta syntymäpäivät täytyi viettää etätapahtumana, emmekä myöskään voineet järjestää omia avajaisiaan uudelle Suomalaisen pelijournalismin mikrokivikausi -näyttelylle. Niinpä asioita päätettiin nivoa yhteen. Jospa avajaisista tehdäänkin osa syntymäpäiväohjelmaa, ja siihen tuodaan lisää pelijournalismihistoriaa. Näin toimittiin.

Pelijournalisminäyttely oli minulle henkilökohtaisestikin tärkeä, vaikka sen kuratointi ja suunnittelu olikin muissa käsissä. Vanhan koulun pelijournalismista ja sen muutoksesta näet riittää kerrottavaa. Se huomattiin myös nettisynttärien tärkeimmässä ja harvinaisimmassa ohjelmanumerossa, kun etäseminaariin saatiin haastateltavakseni kolme henkilöä suomalaisen pelilehdistön taipaleelta.

Tuija Lindén oli mukana tekemässä legendaarisia paperimedioita jo vuonna kivi, jo ennen MikroBitti-lehden perustamista. Helsinkiläinen Tecnopress oli pieni yritys, joka julkaisi muun muassa *Tietokone*-, *Prosessori*- ja *Hifi*-lehtiä. Sen **Lauri Kotilainen** päätti perustaa uuden lehden, *MikroBitin*.

Tuija muistelee alkuaikoja: ”Lauri päätti ottaa minut mukaan. Peliarvostelujahan piti ilman muuta tehdä, joten aloitimme yhteistyön pienen Toptronics-maahantuojan kanssa, että saadaan pelejä arvosteltavaksi. Arvostelut olivat alkuun kahden, kolmen kappaleen kokoisia eikä niissä ollut edes kuvia. Arvosteluja teki sekalainen seurakunta.”

MikroBitin toimitussihteeri Tuija pyöritti pian myös Tecnopressin seuraavaa lehteä, Commodore-koneisiin keskittynyttä *C=lehteä*. Suurin urakka oli vasta edessä: *C=lehden* hiipuessa Tuija ja kumppanit loivat sille korvaajan, peleihin keskittyneen lehden nimeltä *Pelit*.

Petri Teittinen liittyi *MikroBitin* ja sittemmin *C=lehden* ja *Pelitin* leipiin vuonna 1986. Petri kertoo omasta toimittajan uran alkutaipaleestaan, joka on ensimmäisen sukupolven pelitoimittajille monella tavoin perinteinen: alalle päädyttiin kaverien kautta tai yllyttämänä. Porin lähellä Nakkilassa asunut Petri oli näet jäänyt koukkuun videopelikoneisiin jo 1970-luvun puolella, perheen ensimmäisen tv-pelin myötä. Sittemmin Petri törmäsi ja tutustui porilaisessa peli- ja tietokonekaupassa hahmoon, joka vaikutti merkittävästi hänenkin tulevaisuutensa. Pelikaupan myyjä, joku **Niko Nirvi**, kertoi myyneensä peliarvostelun MikroBittiin, josta Petri kävi miettimään, onnistuiko se häneltäkin. Ja onnistuihan se. Juttu Firebirdin *Empire!* -pelistä julkaistiin ja herrasta tuli bittilehtien vakiokasvo. Petri muistetaan erityisen hyvin **Baron Knightlore** -aliaksestaan ja Pelikopteri-uutispalstasta.

”Niko muuten höpläytti minua saman tien: silloin ei ollut käytettävissä nettiä eikä ollut pelilehtiäkään. Kysyin Nikolta, että kirjoitanko arvostelun minä- vai me-muodossa, onko mielipide minun vai lehden. Niko sanoi, että me. Vasta jälkikäteen kävi ilmi, että totta kai se oli höpläytys”, Petri muistelee.

Kaj ”Kaizu” Laaksonen tuli hänkin alalle oudosta suunnasta. Kaizu oli aikoinaan kouluttautunut rakennuspiirtäjäksi, mutta sitten hän jäi työttömäksi. Mitäpä muutakaan työttömänä tekisi kuin pelaisi ja roikkuisi sen aikaisessa internetissä ja keskustelupalstoilla – muun muassa Pelit-lehden foorumilla. Aktiivinen ja foorumeilla hyvin sanan säilää käyttävä kaveri noteerattiin toimituksessa, ja NNirvi kalasti Kaizun mukaan *Pelit*-lehden avustajapiiriin. Siitä urkenikin uusi ura. ”Ei ole enää paljoa taloja piirrelty”, Kaizu kuittaa.

Printistä Pelit-lehteen

Suomalaisen pelijournalismin historia liittyy tiukasti myös suomalaisen tietokonelehdistön historiaan ja nimenomaan Tecnopress-kustannusyhtiöön. Puhtaasti ammattilaisille suunnattujen lehtien lisäksi Suomessa ilmestyi 1980-luvun puolivälin jälkeen kolme merkittävää lehteä, joissa käsiteltiin pelejä: *Printti*, *MikroBitti* ja *C=lehti*. *Printti* ja *MikroBitti* saivat molemmat alkunsa vuonna 1984, mutta *Printin* taival päättyi vuonna 1987. Kaksi kertaa kuukaudessa ilmestynyt sanomalehtimäinen *Printti* oli kuin aikansa sähkösanomauutiset, täynnä liki reaaliaikaista, tuoretta tietoa. Se ei kuitenkaan riittänyt, kaupalliset tavoitteet eivät täyttyneet.

Kuukausittainen *MikroBitti* sen sijaan voi hyvin. Markkinoilla riitti vielä hyödyntämätöntä potentiaalia. Voitaisiinko tasavallan tietokoneen ja sen seuraajan käyttäjät houkutella ulkomaisten lehtien parista kotimaisen vaihtoehdon pariin? Tecnopress perusti erityisen Commodore-koneisiin keskittyvän *C=lehden*, jonka Tuija Lindén otti harteilleen. *C=lehti* sisälsi sekin peliarvosteluja, mutta niille annettiin paikoin enemmän tilaa kuin *MikroBitissä*. *Bitissä* oli asiaa paljon laveammin, monelta alustalta ja koodijuttuakin – niille tarvittiin rutkasti tilaa. *C=*:ssä oli varaa olla vapaampi sivukäytön suhteen.

C=lehti löysi lukijansa ja lukijat sen peliarvostelut. Tecnopressin lehtien päätoimittaja **Eskoensio Pipatti** havaitsi niiden suuren suosion ja päätti kapitalisoida tilanteen: kootaan lehdissä ilmestyneet peliarvostelut vuosikirjoiksi ja höystetään niitä uusilla, yksinomaan kirjoihin teetetyillä erikoisartikkeleilla.

Pelit-kirjat osoittautuivat suuriksi hiteiksi ja tärkeiksi niin kirjojen tekijöille kuin lukijoille. Muistan itsekin kirjojen tekoprosessin olleen kiinnostavaa ja palkitsevaa työtä. Kirjat toimivat myös tehokkaana ylivuotokanavana lehdille: kaikki arvosteltavat pelit eivät mahtuneet millään kahdenkaan eri lehden yhteensä 16 vuosittaiseen numeroon. Niinpä kirjoihin tilattiin varta vasten omia arvostelujaan – eli toisin kuin moni usein luulee, kirjat eivät olleet pelkästään *Bitin* ja *C=*:n arvostelujen kokoelmia. Niihin tehtiin myös tavallista suurempia läpipeluuohjeita merkkipeleistä. Läpipeluuohjeet eivät ole enää tänään niin kuranttia sisältöä, mutta muun muassa Teittisen, Nirvin ja **Jyrki J. J. Kasvin** tekemät messureportaasit, haastattelut ja laajat lukujutut ovat tänään luettuna arvokkaita aikakoneita menneisyyteen. Pitkällä tähtäimellä voinee veikata, että suuret läpipeluuohjeet jäivät myös toimitustiimin mieleen, sillä sittemmin ensimmäisen Pelit-lehden *Monkey Island* -läpäri oli jotain aivan uudenlaista.

Pelit-kirjoja tehtiin kaikkiaan kahdeksan kappaletta. PC:n noustessa kelvolliseksi pelikoneeksi sai DOS-alusta omat kirjansa: *PC-Pelit* 1990 ja 1991.

”Me alettiin tekemään näitä kirjoja Nikon kanssa ja laajensimme niitä artikkeleilla. Muistaakseni esimerkiksi Petri kirjoitti artikkelin **Arnold Schwarzenegger** -peleistä. Niistä tuli niin suosittuja, että ryhdyimme tekemään niitä kaksi vuodessa ja lisättiin vielä PC-versio. Sitten Commodoren maahantuoja PCI-Data meni konkurssiin. Meillä oli heidän kanssaan yhteistyökuvio, jossa saimme heiltä yhteystietoja, jonne pystyimme markkinoimaan *C=lehteä*. Se hetki oli se, kun vuonna 1992 päätimme, että *C=* saa mennä ja perustetaan sen tilalle *Pelit*. Sitä ei varsinaisesti koskaan edes perustettu – kirjat vain laajennettiin ilmestymään ensin kahdeksan, sitten 11 kertaa vuodessa”, Tuija muistelee.

Vaikka esimerkiksi *Printissä* ja ties kuinka monissa paikallisemmissa ja alueellisissa sanomalehdissä oli pelisisältöä, se ei ollut yhtä loogista, yhtä syvälle kurottuvaa eikä etenkään yhtä kattavaa kuin pitkäikäisissä tietokone- ja pelilehdissä. Ei siis ihme, että suomalaisen pelimedian esihistoriassa nimenomaan Tecnopressin lehtien kautta alalle saapui pysyvästi henkilöitä, jotka päätyivät kirjoittamaan peleistä joko ammatikseen tai ainakin freelancereina pitkällä aikavälillä.

Peleistä on kirjoitettu kauan ja sen myötä Suomeen syntyi omanlaisensa, pitkään pieni pelitoimittajien ammattikunta. Ammattia ei ole kuitenkaan juuri tutkittu tai käsitelty, sitä on vain tehty. Ympärillä maailma ja pelaaminen ovat muuttuneet, ja tänään aikuisetkin, jopa yli 30-vuotiaat, voivat pelata videopelejä aivan avoimesti. Myös ammatti on muuttunut. Lehtien määrä on pudonnut rajusti ja painettu media on muuttunut enemmänkin poikkeukseksi, kun julkaisut ovat siirtyneet verkkomaailmaan.

Suomen pelimuseon syntymäpäivälähetyksen valittuja paloja julkaistaan ajan mittaan museon Youtube-kanavalla. Päivän aiheita käsitellään jatkossa muillakin tavoin *Skrollin*, *Retro Rewindin* ja *V2.fi:n* puolella. Suomalaisen pelijournalismin mikrokivikausi -näyttelyyn voi tutustua Suomen pelimuseossa 1.5.2022 saakka.

Jukka O. Kauppinen alias Grendel/Byterapers on pitkän linjan toimittaja, joka kirjoitti ensimmäisiä tekstejään mekaanisella kirjoituskoneella. JOKin ensimmäinen peliarvostelu julkaistiin 1986, mistä lähtien hän on ahkeroinut tauotta kirjoittaen juttuja muun muassa videopeleistä, tietotekniikasta, viihteestä, ilmailusta ja burleskista. JOKstoriaa-sarjassa pureskellaan niin kotimikroilun kuin peli- ja digijournalismin maailmaa grennulasien läpi nähtynä. jukka@skrolli.fi

SKROLLI 6,4 VUOTTA SITTEN

Janne Sirén

Tutustumme *Skrolli 6,4 vuotta sitten* -palstalla Skrolli-lehden historiaan. Nyt vuorossa on Skrollin kolmannen varsinaisen vuosikerran kolmas numero. Olipa kerran synkkä ja myrskyisä yö, tarkalleen 6,4 vuotta sitten...

Skrolli.fi uudistuu

Kesällä 2015 Skrollin toimitus työsti kovaa kyytiä Skrolli-lehden numeroa 2015.3 ja osallistui Assembly Summer 2015 -tapahtumaan Helsingin messukeskuksessa. Käynnissä oli kuitenkin myös kolmas hanke, jonka perässä on palattava lähes kymmenen vuoden taakse.

Skrolli-projekti käynnistyi syyskuussa 2012 (ks. Skrolli 2015.1, s. 72–74), ja alkuun sen tärkein verkkopresenssi oli IRC-kanavilla, kuten #skrolli ja sitä kolmella päivällä edeltänyt #tritti. 10.9. rekisteröityyn skrolli.fi-osoitteeseen saatiin pian aikaiseksi myös staattinen kotisivun alku. Staattisen sivun korvasi toimituksen suunnittelema ja **Erno Kolisevan** toteuttama Drupal-pohjainen sivusto 19.10.2012 – sopivasti ennen seuraavan päivän WÄRK:fest 2012 -tapahtumaa, jossa Skrollin näytenumero 2012.0 esiteltäisiin suurelle yleisölle.

Yleisempää WordPress-sivustomoottoria pidettiin alkuun muun muassa liian blogimaisena Skrollille. Lisäksi Drupal oli tuolloisille tekijöille tutumpi. WordPress-sivustomoottoriin siirtyminen herätti kuitenkin Skrollin toimituksen keskuudessa säännöllistä keskustelua. WP-sivuston meille toteutti lopulta turkulainen **doop**-mainostoimisto Skrolli-projektiin sopivassa talkoohengessä vuonna 2015. Uusi sivusto julkaistiin 31. heinäkuuta eli hieman ennen Skrollin numeroa 2015.3. Sittemmin olemme ylläpitäneet tätä sivustoa itse, ja se palvelee hyvin tänä päivänäkin.

Skrollin web-hotellikumppanina toimi tässä vaiheessa **Capnova** ja toimii edelleen (joskin yrityksen nimi on elänyt). Koko Skrollin historian ajan erilaisten toimijoiden – niin yksilöiden kuin yritystenkin – halukkuus heittäytyä tukemaan projektia on ollut ilahduttavaa ja arvokasta. Suuri kiitos!

Kuumat otot

Skrolli-lehden numero 2015.3 ilmestyi 15. syyskuuta 2015. Edellisen numeron kohdalla menestyksellä alkanut irtonumeromyynti Jimm's-myymälässä Turussa kyykkäsi tosin heti alkuunsa, sillä saimme toimitettua sinne lehteä vasta marraskuussa, vaikka asian piti olla hoidossa. Sittemmin ryhdistäydyimme tässä, mutta muuten sisäinen iskulauseemme pätee edelleen: *Meno ihan Skrolli.*

Olkaapa hyvät, kolme otetta Skrollista 2015.3:

"Nintendonyyttärit, Darrabingo, LOVE-69 Deittailutekniikan perusteet, TEDxOtaniemi, open mic -ilta sekä VJ-työpaja. Nämä ja monet muut tapahtumat hellivät eri alojen opiskelijoita ja kaupunkilaisia Otaniemi pop up -olohuoneessa." – **Elli Kurkikangas**, *Teekkareiden ja taideopiskelijoiden yhteinen olohuone*

"Digitaalisen aineiston säilyttämiskäytännöt ja arkistointi ovat viime aikoina Suomessakin esille nousseita kysymyksiä. Miten michiganilaisen yliopiston videopeli- ja tietokonearkistoa ylläpidetään, hoidetaan ja suojellaan?" – **Mikko Tuomela**, *Pelejä pelastamassa – Haastattelussa videopeliarkiston katastrofisuunnittelija Rebecca Frank*

"Ledi on diodi, jonka voit kuvitella johdonpätkäksi, jonka läpi virta pääsee vain yhteen suuntaan. Jos kytket ledin plus- ja miinusnavan väliin virtalähteeseen toisinpäin, mitään ei tapahdu, koska virta ei kulje. Päinvastoin kytkemällä saat aikaan oikosulun, jolloin ledi vilkahtaa viimeisen kerran. Tätä ei kannata kokeilla tarkoituksella, sillä virtalähde ei välttämättä pidä oikosulusta." – **Ville Ranki**, *Yhden illan projekti: Joystick-testeri*

Skrolli 2015.3:n työstämisen aikaan saimme myös sähköpostia pelitoimittajaveteraanilta ja *Penis Attack* -pelin kehittäjältä **Niko Nirviltä**. Nnirvi lähetti hyvän artikkeli-idean, joka tunnetaan edelleen Skrollin toimituksessa ikuisuusprojektina nimeltään dinosaurusliite. Joku päivä?

Skrolli 2021.4:n 6,4 vuotta sitten -palstalla oli sivumäärävirhe, oikaisu s. 84. Skrolli 2015.3:n pdf-versio vapautettiin maksuttomaan jakeluun 14.12.2015 ja sen voi ladata osoitteesta skrolli.fi/numerot. Skrolli 2015.3 ja useat vanhat irtonumeromme ovat edelleen myynnissä myös paperilehtenä: skrolli.fi/kauppa. Lisäksi vanhoja Skrolleja voi lukea maksutta Internet Archivessa: archive.org/details/skrolli.

Sirén-muisto: Skrolli 2015.3 on merkityksellinen allekirjoittaneelle. Se on ensimmäinen Skrolli, jonka tekoon osallistuin lehden toimituksesta käsin. Olin toki ollut mukana Skrolli ry:n toiminnassa jo vuoden 2013 perustamiskokouksesta lähtien, aloitin kirjoittamisen numerossa 2014.2 ja vastasin esimerkiksi Skrollin irtonumeromyynnin avauksesta numerosta 2015.2 alkaen yhdistyksen hallituksessa. Kesällä 2015 liityin kuitenkin myös Skrollin noin kymmenhenkiseen toimitustiimiin, joka koostaa alati vaihtelevien avustajien kontribuutioista ja omista tekeleistään itse lehden.

Rehellisyyden nimissä vaikutukseni Skrolliin 2015.3 oli kolumniani lukuun ottamatta minimaalinen. Ehkä siksi lehti on niin hyvä. Lähinnä olin osaltani myötävaikuttamassa toimituksen siirtymiseen alun puhtaasta IRC-yhteydenpidosta Slack-palveluun, koska en osannut enää käyttää IRCiä. Skrolli 2015.3 oli ensimmäinen Skrolli, jonka toimitustyössä käytettiin Slack-IRC-yhdistelmää. (Käytämme tätä yhdistelmää edelleen vuonna 2022, enkä vieläkään osaa käyttää IRCiä.) Skrollista 2015.4 alkaen osallistuisin sitten tiukasti myös lehden pilaamiseen.

Kuva: Tero Heino

Skrolli osallistui elokuussa 2015 kolmatta kertaa kesä-Assemblyyn näytteilleasettajana (kuvassa) ja mm. arpoi tapahtumassa keskipöydällä näkyvän Commodore 64 -tietokoneen. Vuosien 2013 ja 2014 peruspöytien sijaan Skrollin ympärille alkoi hahmottua oma tapahtumaosastonsa. Kokonaisuus kasvaisi tästä lähtien vuosi vuodelta, huipentuen koronaa edeltäneeseen Skrolli-lavaan kesän 2019 Assyilla (ks. Skrolli 2019.3, s. 86–87).

Kuva: Toni Kuokkanen

Skrollin Assembly-osaston vetonaulana oli tuolloin 30 vuotta täyttänyt Commodore Amiga 1000 -kotitietokone. Skrollissa muistetaan osastolla ympäripäissään käynyt tapahtumavieras, joka halusi käyttää tätä Amigaa levykkeidensä formatointiin ja läpikäyntiin. Erittäin liikuttuneesta tilastaan huolimatta Amigan käyttöjärjestelmä tuli vieraalle yhä selkäytimestä – kaveri viihtyi paikalla pitkän tovin ja kirjoitti lennosta jopa AmigaDOS-skriptin tarpeisiinsa.

TROLLI 2015.1
VALINNAISEN TIETOTEKNIIKAN ERIKOISLEHTI

Skrollin Facebook-sivulla (facebook.com/skrollilehti) kerrottiin kesällä 2015 koulun tietotekniikkaryhmässä syntyneestä lehdestä, josta julkaistiin tekijän luvalla otos: "Nykyajan nuoretkin osaavat: rovaniemeläiset koululaiset tekivät oman Trolli-lehden. (Logo: Markus Kallatsa)" Logon lisäksi kannessa oli internetistä otettua kuvamateriaalia, joten koko kantta emme uskaltaneet julkaista; tuotokseen sisältyi kuulemamme mukaan myös kokonainen lehti. Trolli-lehti-vitsiä on sittemmin kuultu Skrolliin liittyen säännöllisesti, viimeksi Skrolli 2021.4:n Postipalstalla. Muun muassa artikkelien loppumerkkinä esiintyvä maskottimme, Skrolli-trolli, on toki trolli-huomiosta aina hyvillään.

Pelinkehityksen vallankumous:
ammattityökaluja ilmaiseksi

Eek-O-Mittari: 79 %

Skrolli pyrkii kestävämpään journalismiin. Eek-O-Mittarissa arvioimme, kuinka suuren osan vanhasta lehdestä voisimme painaa uudestaan.

Prosenttilukuun päädyttiin arvioimalla, että Skrolli 2015.3:n 70:stä mainoksettomasta sivusta yhteensä noin 11 + 4 sivun edestä sisältöä on vanhentunut. Loput voisi jotakuinkin julkaista edelleen.

Edeltävän numeron joka toinen sivu tuntui vanhentuneelta, mutta 2015.3:ssa juuri mikään ei haiskahda ihan heti. Juttuvalikoimassa on hitaasti liikkuvia ja historiallisia aiheita. Poikkeuksena *Neuroverkkojen mysteerit auki* käsittelee nopealiikkeisen tekoälyteknologian yksityiskohtia. Toinen haaste on *Pelinkehityksen vallankumous* – sen ydinasetelma ei ole muuttunut, mutta modernissa uusinnassa käsiteltäisiin arvatenkin myös uusimpia avoimia vaihtoehtoja sekä päivitettäisiin useita kohtia. Molemmat artikkelit ovat rajatapauksia, mutta laskettiin lopulta vanhentuneiksi.

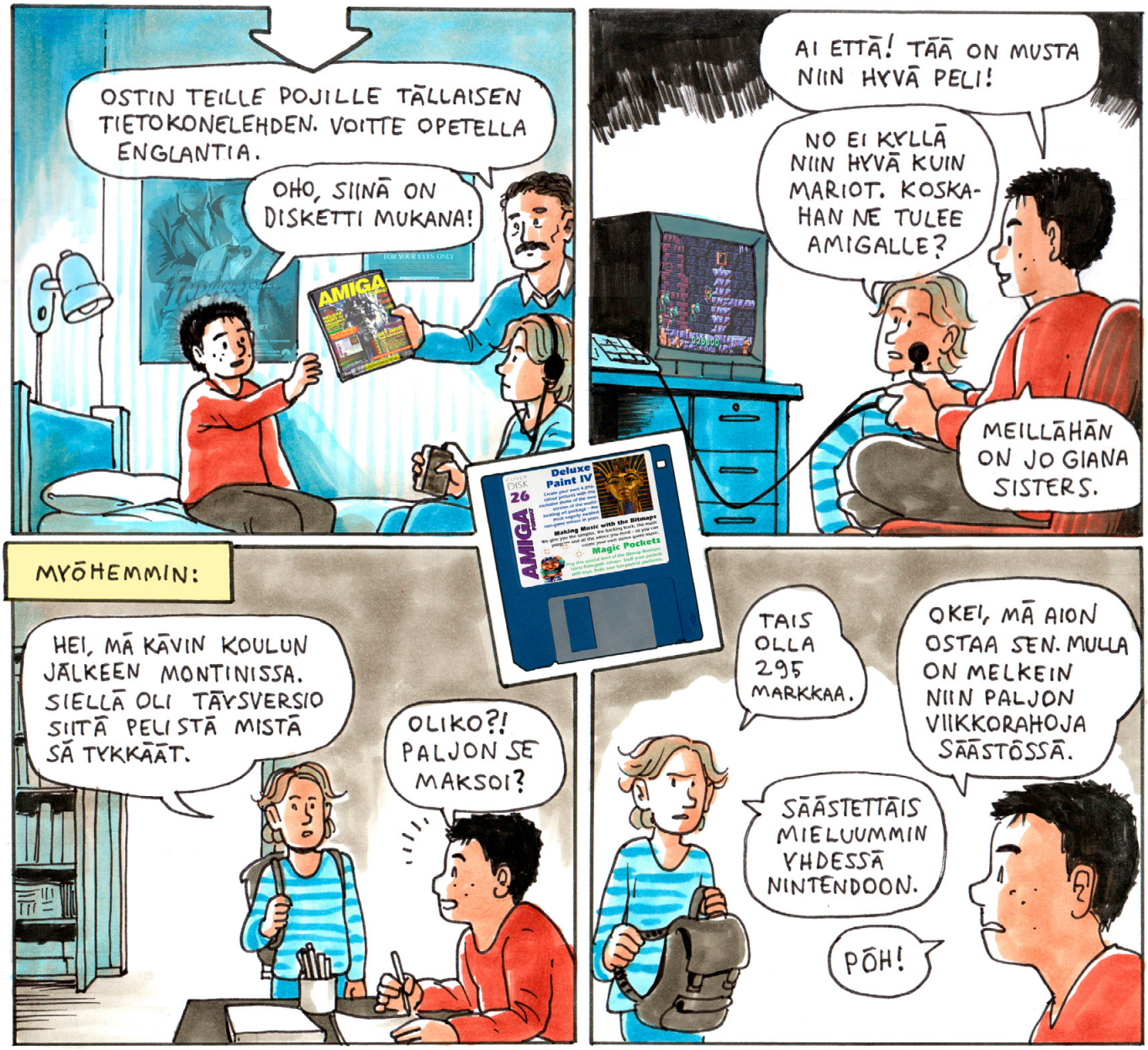
Lisäksi *MenuetOS*, *Vi-editorien historia*, *Maasta Kuuhun*, *Lisää tallennustilaa Amigaan* sekä *Everdrive* sisältävät hieman uusittavia tuoteviittauksia, vaikka itse jutut toimivatkin edelleen. *Tietokone typografian tuhoajana*, *Nokian paluu* ja *Satunnainen samooja* ansaitsisivat päivitetyt lopetukset. Näistä verotettiin puoli sivua kustakin.

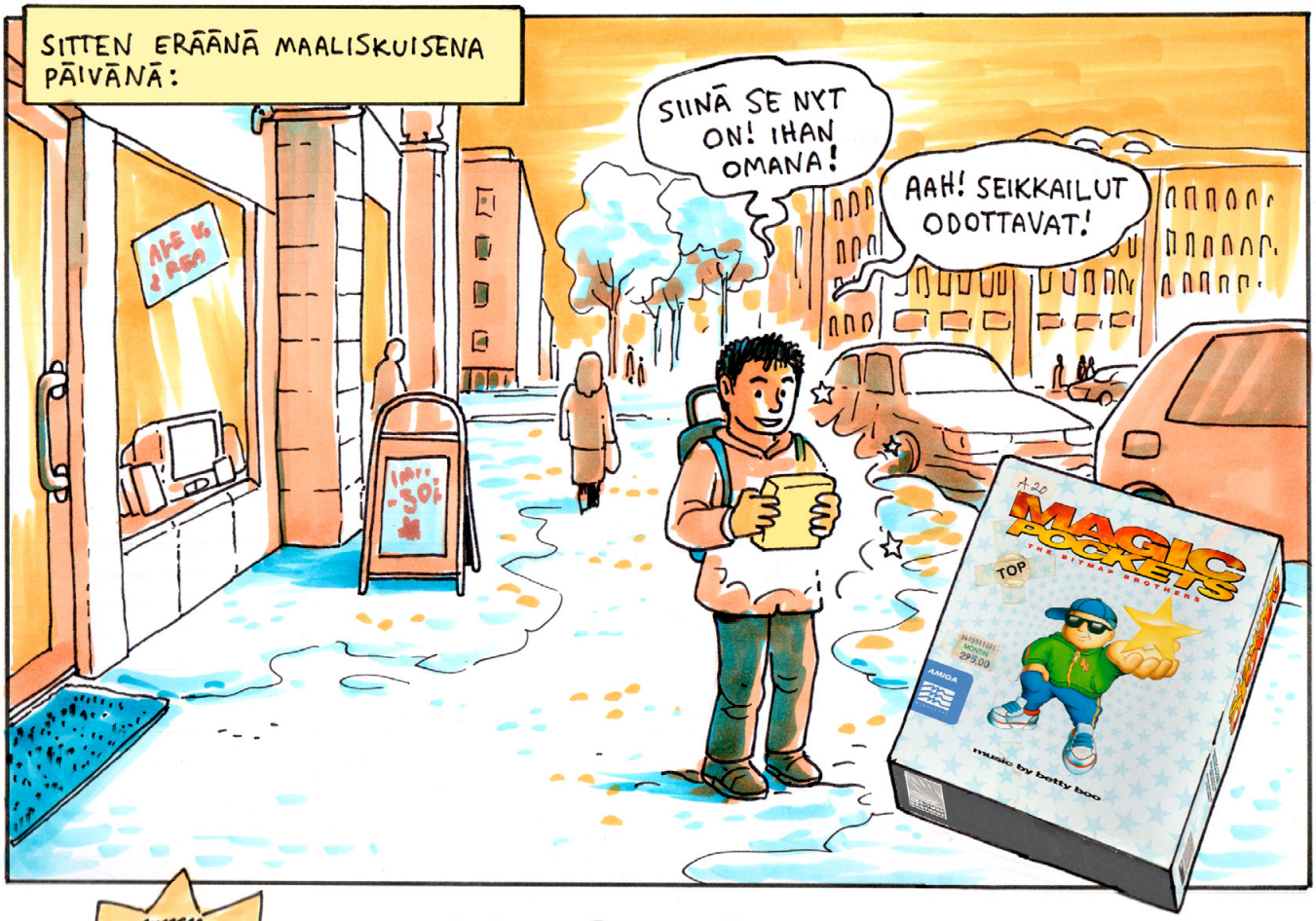
Lehdessä on toki myös pari haastattelu- ja tapahtumajuttua, jotka ovat ajassaan kiinni. Ihmisiä ja kulttuuritapahtumia koskevat reportaasit tuntuvat kuitenkin kestävän aikaa teknologiajuttuja paremmin. Monet niistä voisi painaa uudestaan inhimillisesti kiinnostavina historia-artikkeleina vain lisäämällä vuosiluvun otsikkoon – sama temppu tuskin toimisi pari vuotta vanhan näytönohjainvertailun kohdalla (tai ehkä komponenttipulan seurauksena toimisikin).

IKUISUUSPROJEKTI

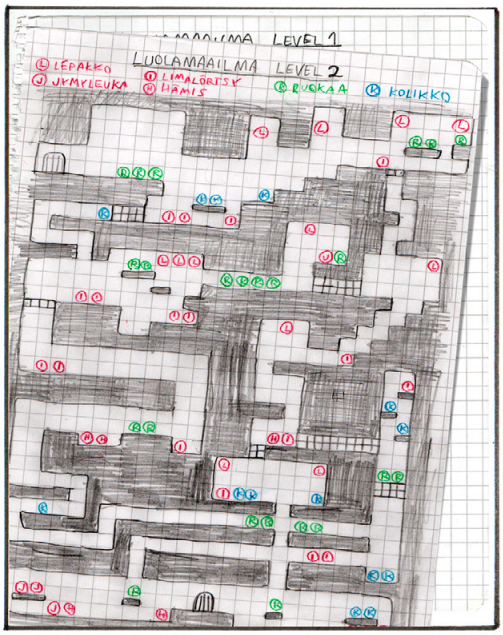
JJNÄÄS22
Jussi Pakkanen

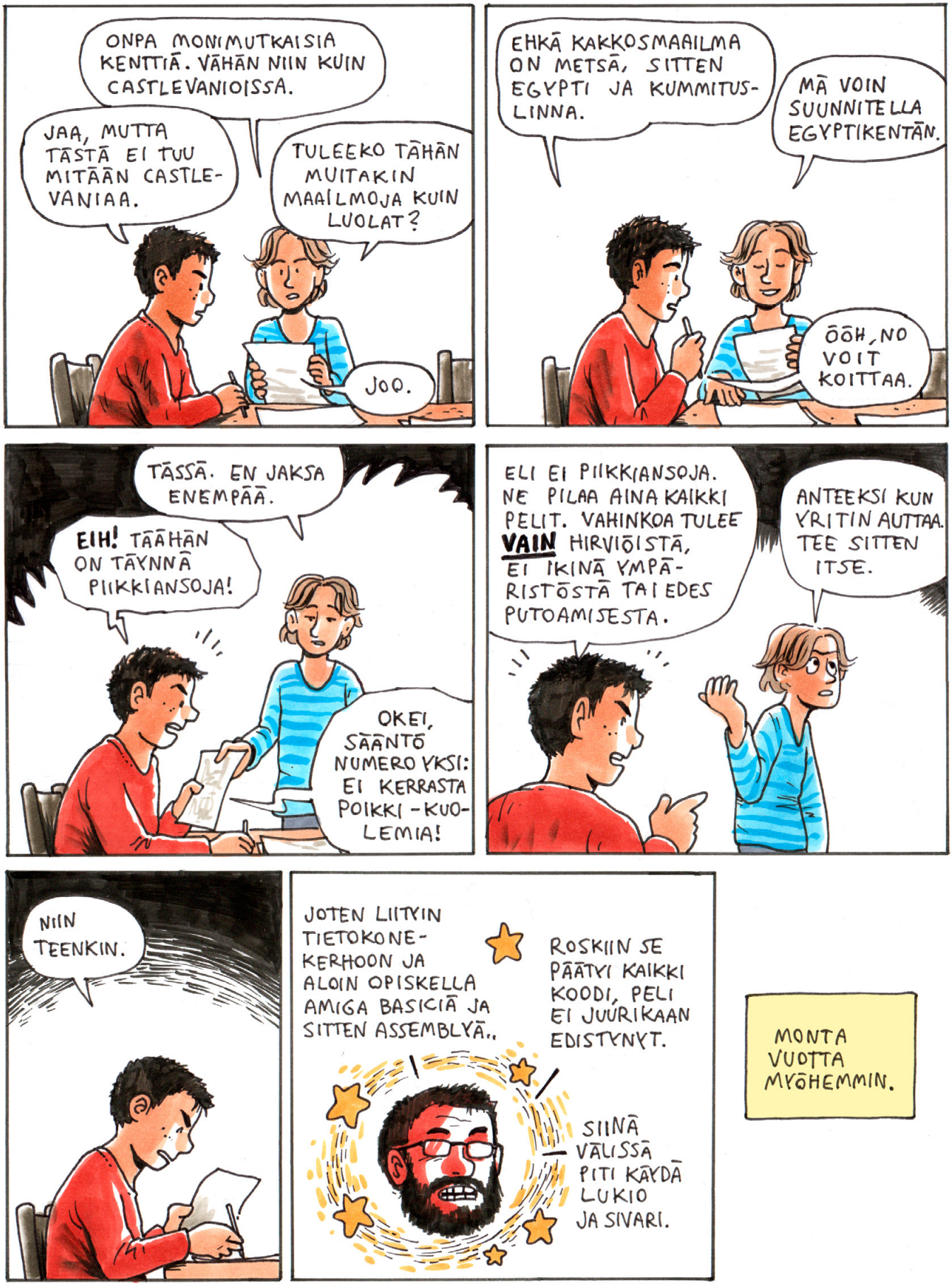
Hypätään vuoteen 1991





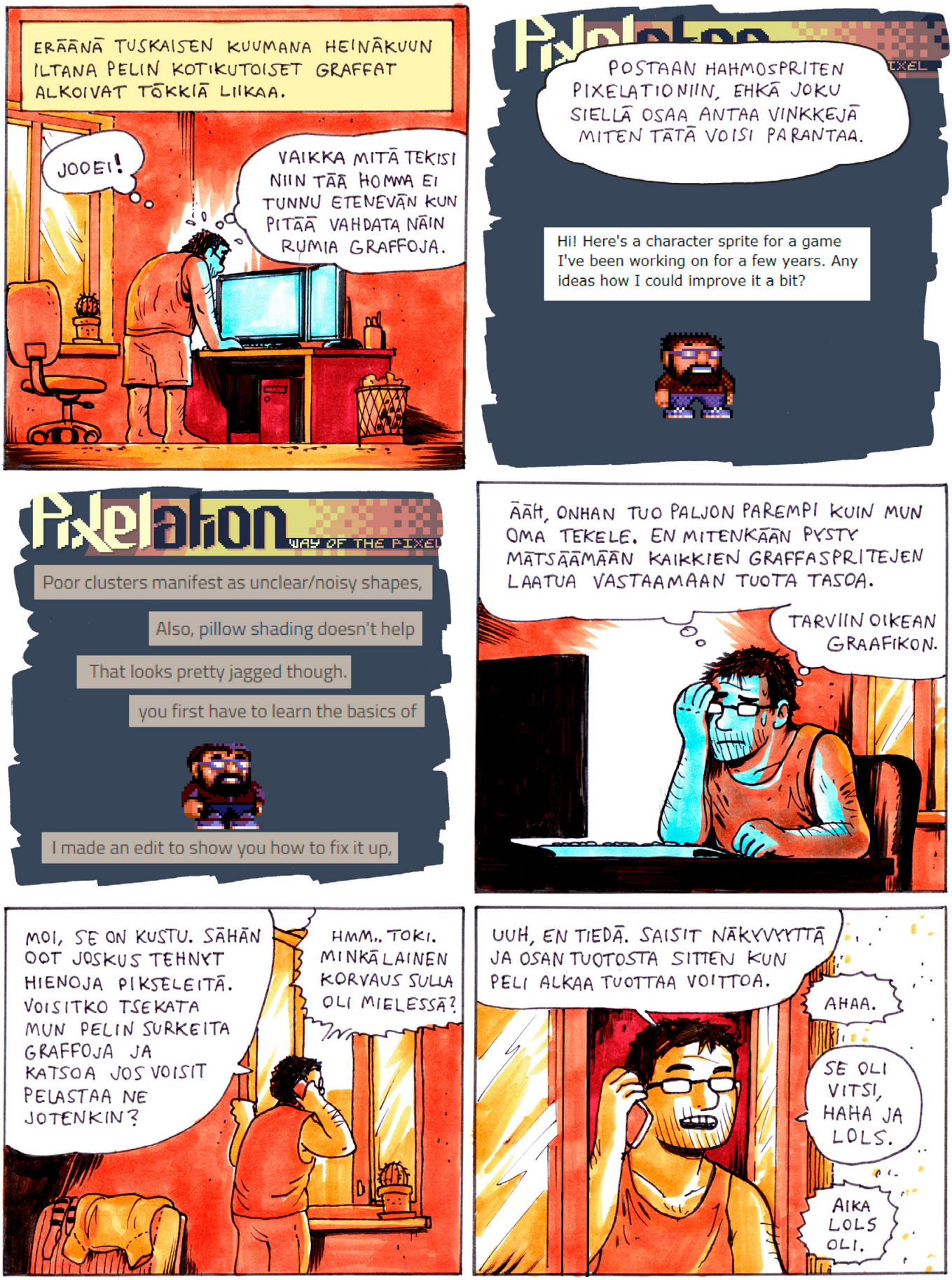
ONHAN SIINÄ PELISSÄ KAIKENLAISTA PIKKUVIKAA, JOSTA JOTKUT AINA NILLITTÄVÄT PELATTUAAN SITÄ VARTIN. NO JOO, NYKIVÄ VIERITYS, TAHMEAT KONTROLLIT, JA KOLMAS JA NELJÄS MAAILMA OVAT HIEMAN LAIMEITA KAHTEEN EKAAN VERRATTUNA, MUTTA TYKKÄSIN SILTI SIITÄ NIIN PALJON, ETTÄ PÄÄTIN...



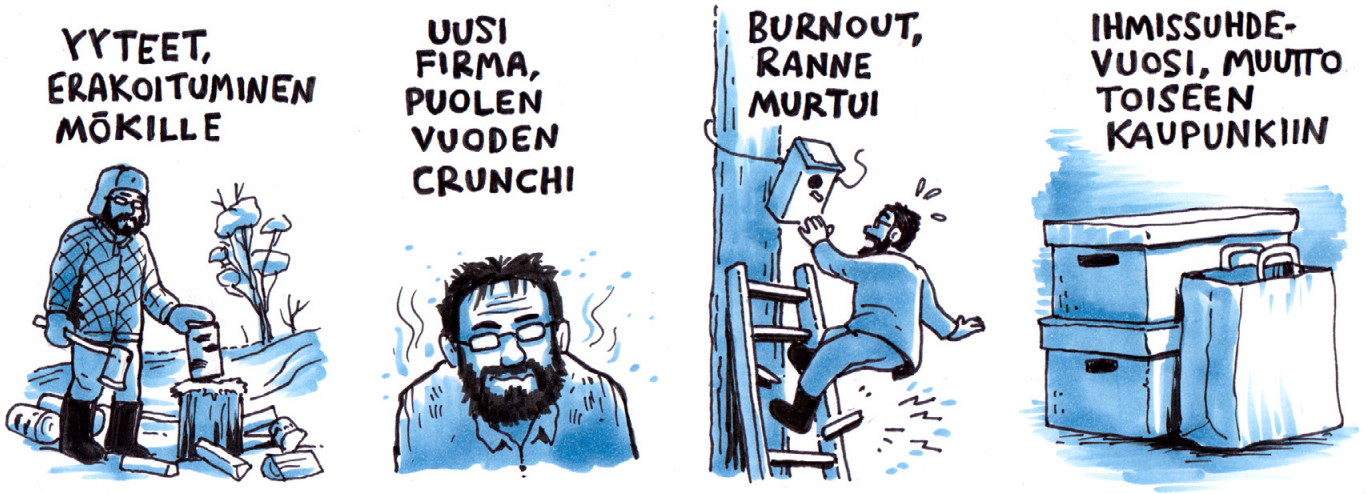

ONPA MONIMUTKAISIA KENTTIÄ. VÄHÄN NIIN KUIN CASTLEVANIOISSA.
JAA, MUTTA TÄSTÄ EI TUU MITÄÄN CASTLEVANIAA.
TULEEKO TÄHÄN MUITAKIN MAAILMOJA KUIN LUOLAT?
JOO.
EHKÄ KAKKOSMAAILMA ON METSÄ, SITTEN EGYPTI JA KUMMITUSLINNA.
MÄ VOIN SUUNNITELLA EGYPTIKENTÄN.
ÖÖH, NO VOIT KOITTAA.
TÄSSÄ. EN JAKSA ENEMPÄÄ.
EIH! TÄÄHÄN ON TÄYNNÄ PIIKKIANSOJA!
OKEI, SÄÄNTÖ NUMERO YKSI: EI KERRASTA POIKKI -KUOLEMIA!
ELI EI PIIKKIANSOJA. NE PILAA AINA KAIKKI PELIT. VAHINKOA TULEE VAIN HIRVIÖISTÄ, EI IKINÄ YMPÄRISTÖSTÄ TAI EDES PUTOAMISESTA.
ANTEEKSI KUN YRITIN AUTTAA. TEE SITTEN ITSE.
NIIN TEENKIN.
JOTEN LIITYIN TIETOKONEKERHOON JA ALOIN OPISKELLA AMIGA BASICIÄ JA SITTEN ASSEMBLYÄ..
ROSKIIN SE PÄÄTYI KAIKKI KOODI, PELI EI JUURIKAAN EDISTYNYT.
SIINÄ VÄLISSÄ PITI KÄYDÄ LUKIO JA SIVARI.
MONTA VUOTTA MYÖHEMMIN.

MUUTIN TOISEEN KAUPUNKIIN OPISKELEMAAN, JA PAKATESSANI KAMOJA LÖYSIN KAIKKI VANHAT SUUNNITELMANI JA MUISTIINPANONI.
AI NIIN, TÄMÄKIN JUTTU...
KYLLÄHÄN TÄSSÄ OLI PALJON HYVIÄKIN IDEOITA.
PITÄISKÖ YRITTÄÄ HERÄTTÄÄ PROJEKTI HENKIIN.. ALOITTAA IHAN ALUSTA..
JOS MÄ EN TEE TÄTÄ NIIN EI TEE KUKAAN MUUKAAN.
MINULLA OLI AVAINKORTTI JOLLA OLI 24/7 PÄÄSY OPISTOLLE, JA VEDIN SIELLÄ MONTA ALLNIGHTERIA KOODATEN PELIÄ OPISTON KONEILLA, KOSKA MINULLA EI OLLUT VARAA OMAAN KONEESEEN.
ILMASTOINNIN HUMINAA
SIRINÄÄ
AAH, KELLO ON KAKSI. TÄNNE EI ENÄÄ VARMASTI TUU KETÄÄN ENNEN AAMUA. IHANA RAUHA.
SIRINÄÄ
VUOSIA KULUI. LOPPUTYÖKIN VIIVÄSTYI, TOSIN EI PELKÄSTÄÄN PELIPROJEKTIN VUOKSI. JOSSAIN VÄLISSÄHÄN PITI MYÖS RYYSTÄÄ TARPEEKSI OLUTTAKIN.
OON VÄHÄN OMAA PELIÄ VÄRKÄNNYT, MUTTA HOMMA ON VÄHÄN JUMISSA OLLUT...
NÄYTÄ SE MULLE JOSKUS KUN TUUN KÄYMÄÄN.
TÄLLANEN. SE ON VÄHÄN KARU VIELÄ.
ONHAN TÄSSÄ.. ÖÖHMM, TYÖSTÄMISKELPOINEN PERUSTA.
SAANKO VILKAISTA SUN KOODIA?

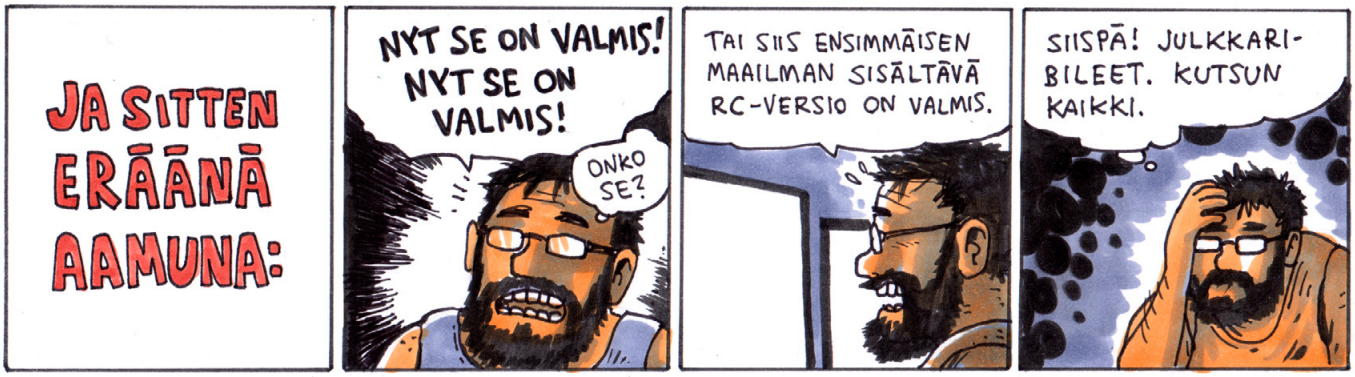
JAAHAS, JOO, AIKA JÄNNÄÄ TAIKAKOODIA. TÄSSÄ ON ONGELMANA SE, ETTÄ KUN YRITÄT RATKOA LIIKKUVIEN ESINEIDEN TÖRMÄYSTÄ PELAAJAAN, HYPPÄÄMISTÄ JONKUN MONSUN PÄÄLTÄ JA SILLEEN... SE ON SUORAAN KOODATTUNA TOSI MONIMUTKAISTA JA BUGIALTISTA.
JOO, NO, EN NYT PYSTYNYT PAREMPAAN.
HALUATKO ETTÄ KOKEILEN PARIA JUTTUA?
NO JOS VIITSIT.
NIIN SIIS JOS HINNASTA SOVITAAN.
KOLME LÖWENBRÄU -TÖLKKIÄ MYÖHEMMIN:
NO NIIN, NYT TÄMÄ TOTEUTUS NÄYTTÄISI TEKEVÄN SAMAT ASIAT.
MITÄ SÄ TEIT? EIHÄN TOSSA OO KUIN JOTAIN KYMMENEN RIVIÄ KOODIA!
KÄYTIN VÄHÄN LISÄÄ MATEMATIIKKAA. ONGELMA REDUSOITUU MINKOWSKI -EROTUKSELLA TAVALLISEKSI VIIVAN JA LAATIKON LEIKKAUKSEN SELVITTÄMISEKSI.
MITÄ SÄ HALUAT PALKKIOKSI?
PULLO JOTAIN SINGLE MALTIA VAIKKA.
PUUH! MUN PITÄÄ VARMAAN SÄÄTÄÄ PALJON MUUTAKIN UUSIKSI. EHKÄ MÄ VOIN PYYTÄÄ LISÄAPUA JOSKUS TOISTEKIN?
KYLLÄ MULLE AINA HYVÄ VISKI MAISTUU.
SITTEN JOUDUIN TÖIHIN JA PROJEKTI UNOHTUI TAAS PITKÄKSI AIKAA.
TUU MEIDÄN FIRMAAN. MÄ VOIN SUOSITELLA SUA.
EN MÄ TIEDÄ... MÄ EN OO MIKÄÄN GURU MISSÄÄN, MUTTA TEEN VÄHÄN KAIKKEA, KESKINKERTAISESTI.
JOS TEILLÄ ON SELLASELLE KÄYTTÖÄ NIIN..

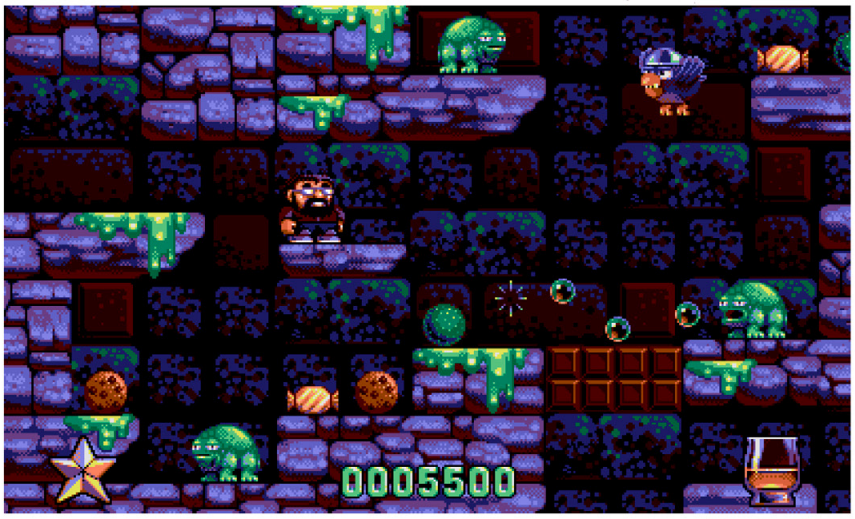

ERÄÄNÄ TUSKAISEN KUUMANA HEINÄKUUN ILTANA PELIN KOTIKUTOISET GRAFFAT ALKOIVAT TÖKKIÄ LIIKAA.
JOOEI!
VAIKKA MITÄ TEKISI NIIN TÄÄ HOMMA EI TUNNU ETENEVÄN KUN PITÄÄ VAHDATA NÄIN RUMIA GRAFFOJA.
POSTAAN HAHMOSPRITEN PIXELATIONIIN, EHKÄ JOKU SIELLÄ OSAA ANTAA VINKKEJÄ MITEN TÄTÄ VOISI PARANTAA.
Hi! Here's a character sprite for a game I've been working on for a few years. Any ideas how I could improve it a bit?
Pixelation
WAY OF THE PIXEL
Poor clusters manifest as unclear/noisy shapes,
Also, pillow shading doesn't help
That looks pretty jagged though.
you first have to learn the basics of
I made an edit to show you how to fix it up,
ÄÄH, ONHAN TUO PALJON PAREMPI KUIN MUN OMA TEKELE. EN MITENKÄÄN PYSTY MÄTSÄÄMÄÄN KAIKKIEN GRAFFASPRITEJEN LAATUA VASTAAMAAN TUOTA TASOA.
TARVIIN OIKEAN GRAAFIKON.
MOI, SE ON KUSTU. SÄHÄN OOT JOSKUS TEHNYT HIENOJA PIKSELEITÄ. VOISITKO TSEKATA MUN PELIN SURKEITA GRAFFOJA JA KATSOA JOS VOISIT PELASTAA NE JOTENKIN?
HMM.. TOKI. MINKÄLAINEN KORVAUS SULLA OLI MIELESSÄ?
UUH, EN TIEDÄ. SAISIT NÄKYVYYTTÄ JA OSAN TUOTOSTA SITTEN KUN PELI ALKAA TUOTTAA VOITTOA.
AHAA.
SE OLI VITSI, HAHA JA LOLS.
AIKA LOLS OLI.

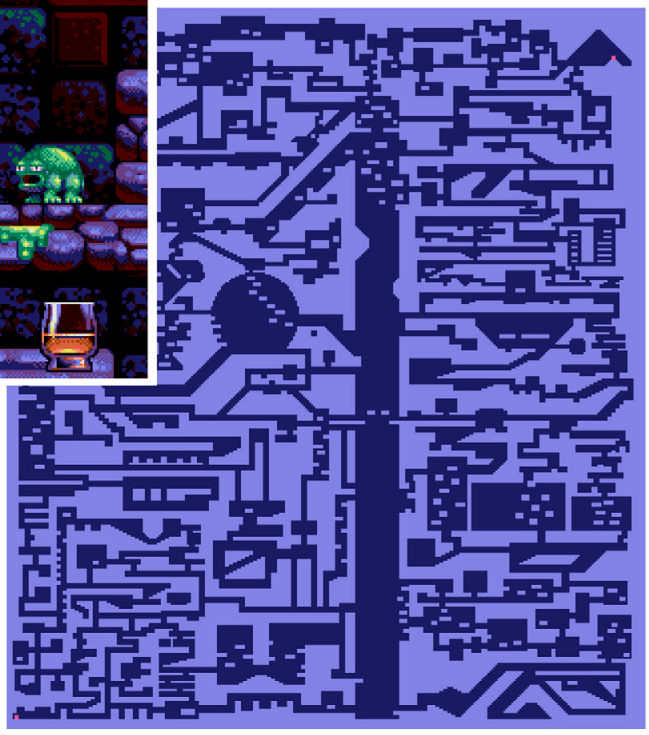
SITTEN TAPAHTUI TAAS KAIKENLAISTA:



Takaisin siihen hetkeen mistä tämä juttu alkoi:







JULKISTUSTILAISUUS TALOYHTIÖN KERHOTILASSA:
NO NIIN, EN OO USKALTANUT LASKEA KUINKA MONTA VUOTTA TÄHÄN ON MENNYT, MUTTA PELI ON NYT VIIMEINKIN SUUNNILLEEN SELLAINEN KUIN OON SEN AINA MIELESSÄNI NÄHNYT.
OTTAKAA JUOMAA JA PELATKAA IHAN RAUHASSA, MÄ PAISTAN VIELÄ LISÄÄ PITSAA. BUGEJA EI TARVITSE RAPORTOIDA, KOSKA NIITÄ EI OLE.
NÄISSÄ KAIKISSA PITSOISSA ON SITTEN ANANASTA.
IHAN HYVÄ TÄSTÄ TULI, KUSTU!
TÄÄ SUN BIISI KUULLOSTAA IHAN JOLTAIN ELWOODIN HITAALTA BIISILTÄ.
SIINÄ TUSKIN LIENEE MITÄÄN VÄÄRÄÄ.
EHDOTIN AIKOINAAN, ETTÄ PITÄISI OLLA PIIKKIANSOJA. VÄHÄN NIITÄ JÄÄ KAIPAAMAAN.
VOIS LIIKKUA VÄHÄN NOPEEMMIN TUO UKKO, JA KENTÄT ENEMMÄN NIIN KUIN JOISSAIN METROIDVANIOISSA.
HYS!
SINÄ YÖNÄ:
PELIT
HUIPPU
98

SKROLLI

TIETOKONEKULTTUURIN ERIKOISLEHTI

+ retro rewind

TILAA SKROLLI KOTIIN TAI KONTTORILLE

Skrollin tilaukset, irtonumerot ja digitaaliset pdf-lehdet verkkokaupasta.
Tilaamalla paperilehden verkosta saat myös pdf-lehden.

tilaaskrolli.fi

B17-klassikkolentelyt

Mikä ihmeen NUON?

Afrikan Tähti

retro rewind

RETROPELAAJAN TIETOPAKETTI 1-2022

MECHAROOLIPELIT

PS1:llä ja PS2:lla

ALAN WAKE

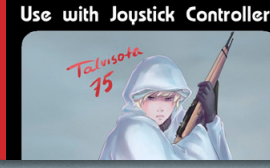
YHÄ MESTARITEOS

SONIC

ei juokse vain Segalla

TALVISOTA

peleissä ja Atari 2600:lle



RuneScape 20 vuotta

Mitä kuuluu MiSTer?

RetroTink ja MClassic

1 / 2022

Päätoimittaja
Jukka O. Kauppinen

Toimitus: Mikko Heinonen, Jukka O. Kauppinen, Manu Pärssinen

Taitto, AD, kuvankäsittely
Manu Pärssinen

Avustajat: Miika Auvinen, Aleksi Vaittinen

Tuotanto: Alasin Media Oy, 1932899-7

Kannen kuva:
Remedy Entertainment

ONKO NYT PEAK RETRO?

Maailma on avautumassa koronapandemian hellittäessä otettaan – ehkä tilapäisesti, ehkä aina seuraavaan pandemiaan asti. Tätä lukiessasi olet asiassa viisaampi kuin me kirjoittajat. Voimme siis jälleen tavata muita ihmisiä, kokoontua Suomen pelimuseoon ja muihin paikkoihin harrastamaan yhdessä ja käydä myös kirpputoreilla metsästämässä vapaammin.

Yksi pandemian vaikutus kuitenkin säilyy vielä pitkään: lähes kaiken hintojen kohoaminen. Etenkin Yhdysvalloissa, jossa koronatukia jaettiin suoraan shekkeinä ihmisille, hyväosaisemmat saivat käytännössä ylimääräistä rahaa postissa. Moni ilmoitti käyttävänsä tämän summan ostaakseen jonkin kauan havittelemansa pelin tai pelikoneen, jonka pyyntihinta oli aiemmin tuntunut liian korkealta. Kuten tunnettua, eBaysta voi ostaa melkein mitä tahansa koska tahansa, jos kukkaron nyörit ovat tarpeeksi löyhällä. Jokainen toteutunut kauppa taas on alustava hintataso seuraavalle. *Retro Rewind* ei ole tähän mennessä kommentoinut politiikkaa eikä aloita sitä nytkään, mutta ilmiö ei tee kokonaisuutena hyvää harrastukselle. Toisaalta myös Suomessa on nähty viime aikoina melko hulvattomia esimerkkejä hintaralleista, kuten *Uuno Turhapuro muuttaa maalle* -pelin uusi hintaennätys 3 712 euroa.

Osaltaan tämä kertoo toki retropelien arvostuksesta, mutta asialla on kääntöpuolensakin. Mitä enemmän raha alkaa ratkaista, sitä enemmän pelien liepeillä alkaa pyöriä porukkaa, jonka motiivi on aivan muu kuin pelihistorian fiilistely ja säilöminen. Ulkomailla varteenotettavia epäilyksiä hintakeinottelusta ja jopa rahanpesusta on jo herännyt. Vielä suurempi uhka koko harrastukselle on kuitenkin se, jos harvinaiset pelien prototyypit ynnä muut historialliset aineistot nousevat hirmuisiin hintoihin ja päätyvät spekuloijien käsiin lukkojen taakse eivätkä museoihin tutkijoiden käsiin ja verkkoon harrastajien iloksi.

Games are meant to be played, not displayed.

Retrosedät Jukka, Manu ja Mikko
retro@retrorewind.fi

SISÄLLYSLUETTELO

107 Ajankohtaisia RETROUUTISIA peleistä ja laitteista
109 Varhaisten Pleikkarien MECHA-ROOLIPELIT
112 ALAN WAKE – kauhutrilleriä ja kuumehoureita
115 Vieläkö maistuu? B-17-klassikkolentelyt
116 Arvostelussa CARRIER COMMAND 2 ja DARKQUEST 2
117 Poimintoja pelimuseosta: AFRIKAN TÄHTI
118 Testissä uusia C64-PELEJÄ
119 Peliarvostelussa PÄRISTELYÄ ja SEIKKAILUA
120 Vieläkö maistuu? GAME BOY, NES ja SNES
122 RETROTINK ja MCLASSIC – parempaa retrokuvaa?
124 Atari 2600 TALVISODASSA – haastattelussa tekijä
126 Peliarvioissa mm. CLOCKWORK AQUARIO
127 SONIC THE HEDGEHOG rynnisti kuusnepalle
128 Erikoisempia SONIC-pelejä
128 Mitä kuuluu MISTER -FPGA-projektille?
131 Testissä Evercaden CODEMASTERS COLLECTION
132 Kirja-arviossa RUNESCAPE: THE FIRST 20 YEARS
133 VUODEN 2021 parhaat pelimusiikkielämykset
134 MANUN KUMMAT: Nuon

INTELLIVISION VAIKEUKSISSA

Intellivision oli yksi 1980-luvun alun paremmista pelikonsoleista – ellei jopa se paras. Laitetta myytiin kaikkiaan kolmisen miljoonaa kappaletta ja sille tehtiin sukupolvensa parhaat ja kauneimmat konsolipelit. Laitteessa siis riittäisi potentiaalia asialliseen uutuuskonsoliin, vaikkapa Intellivision Miniin. Niin uskoi myös Intellivision Entertainment, joka kuitenkin tähtäsi paljon korkeammalle. Ei tehdä Miniä, tehdään uudenlainen viihdealusta, jolla voi tehdä paljon muutakin kuin pelata vanhoja Intellivision-pelejä.

Viimeisimmät käänteet Intellivision Amico -konsolista ovat kuitenkin synkkiä. Toimitusjohtaja ja suurin osakkeenomistaja, pelimuusikkonakin tunnettu **Tommy Tallarico** siirtyi tai siirrettiin syrjään yhtiön johdosta. Julkaisuaikataulu on lipsunut jo kahdella vuodella ja yhtiö on käytännössä kadottanut rahaa kiinalaiseen kaivoon. Yhtiöllä on tyhjä kassa mutta velkaa yli kahdeksan miljoonaa dollaria. Se yrittääkin kerätä lisää varoja toimintaansa neljännellä ja viidennellä rahoituskierroksellaan. Aikaisemmin joukkorahoitukseen sijoittaneiden mahdollisuudet saada vastinetta rahalleen vaikuttavat heikoilta.

Intellivision-yhtiö myöntää itsekin, ettei heillä ole juurikaan osaamista tuotteidensa hinnoittelusta tai markkinoinnista. Tällä ei liene juuri nyt suurtakaan vaikutusta tilanteeseen, koska yhtiöllä ei ole vieläkään valmista tuotetta, jota olisi mahdollista tilata ja saada. Suuria puheita ja yrityksiä pitää yhtiö kiinnostavana sijoittajien silmissä sen sijaan riittää, ja onpa ilmaan heitelty esimerkiksi puheita NFT:nä myytävistä peleistä. Voi olla, että yhtiö joutuu julkaisemaan Amicolle tekemiään pelejä muille alustoille, jos se haluaa edes jonkinlaista liikevaihtoa.

Amicon tilanne muistuttaa siitä, ettei retrohistorian kaupallinen hyödyntäminen ole aina niin helppoa. Hyvä Intellivision Mini varmasti kelpaisi harrastajille AtGamesin Flashback-laitteen tilalle. Amicossa kuitenkin tavoitellaan laajempaa yleisöä, koko perhettä, ja laitteelle kehitetään aivan omia pelejään. Tavalliseen älypuhelintekniikkaan pohjautuva, täysin uusi pelialusta verkkopalveluineen ja latauskauppoineen on kuitenkin aivan eri mittaluokan haaste. Toivottavasti Intellivision Entertainment kuitenkin saa joskus tuotteen myyntiinkin, eikä siitä tule uutta Ouyaa.

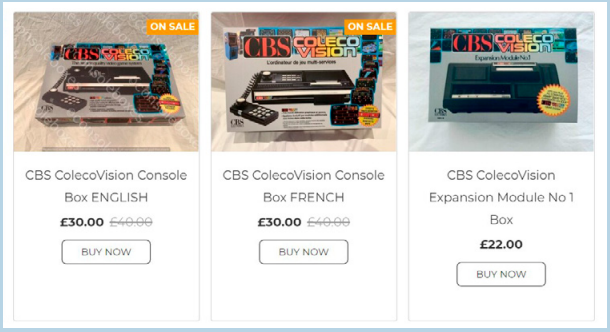
REPROLAATIKOITA RETROKONEILLE

Ostitko Toristä vanhan pelikonsolin tai mikron? Heititkö oman retrolaitteesi laatikon pois 30 vuotta sitten ja nyt kaduttaa? Ei hätää! Uusi palvelu tekee sinulle reproboksejа ja voi että kun hyllynpäällinen näyttää nyt hyvältä.

Palvelu kehuu laatikoidensa olevan erittäin korkealaatuisia ja käyttöä kestäviä painotuotteita, jotka toimitetaan valmiiksi koottuina. Tarjolla on laatikoita muun muassa 3DO:lle, BBC:lle, ColecoVisionille, Commodore 64:lle, VIC20:lle, Dreamcastille, Nintendo 64:lle, Spectrumeille ja Vectrexille. Laatikoista on myös erilaisia versioita.

Reproboksit maksavat 30–45 puntaa plus postit, alvit ja käsittelymaksut. Ei siis aivan halpaa, mutta ilmeisesti hyvää – ja jos kunnon boksilla saa kokoelmansa täydellisemmäksi ja mielihyväreseptorit väpäjämään, niin kyllä se silloin lienee hintansa arvoinen.

Lisätietoja: consoleboxes.com



ASTETTA PAREMPI PS2-MODI

Miten olisi PlayStation 2 -modi, joka ei vaadi sen enempää modaamista kuin tuunaamistakaan? MC2SIO/MX4S-muistikorttiviritys tekee PS2-retroilusta helpompaa kuin koskaan, sillä siinä konsoliin liitetään SD-massamuisti konsolin muistikorttiliitännän kautta.

Uudella ja edullisella SD-ratkaisulla on mahdollista korvata vaivattomasti vanhentuvat mekaaniset kiintolevyt vanhemmissa PS2-malleissa. Slim-konsoleissa taas ei ollut lainkaan laajennuspaikkaa, joten MC2SIO tuo niillekin uutta elämää. Toki kaikissa PS2-malleissa on voinut käyttää myös USB-tikkupohjaista massamuistiratkaisua (esimerkiksi MCBootin kanssa), mutta muistikorttipaikan kautta lataaminen on merkittävästi nopeampaa kuin konsolien USB 1.1 -liitännästä (24 Mbps vs 12 Mbps). Sen kautta saavutetaan riittävä lukunopeus esimerkiksi FMV-videoiden nykimättömään toistamiseen. Pelit on tosin ensin siirrettävä talteen yhteensopivassa formaatissa.

MC2SIO-sovittimen sekä hommaan tarvittavan FreeMCBoot-muistikortin voi tilata osoitteesta heldergametech.com/product-category/ps2/.

Laite on mahdollista rakentaa itsekin – tekniikasta ja ohjelmistosta kiinnostuneet löytävät lisätietoja osoitteesta trisaster.de/page/index.php?topic=575.

DYYNIVATKAIN KAIVETTIIN NAFTALIINISTA

Syystä jos toisesta julkaisematta jääneitä pelejä löytyy aina silloin tällöin. Uuden *Dune*-elokuvan myötä eräs kadonnut ja unohtunut peliprojekti intouduttiin kaivamaan esiin, ja sen kautta myös erään edesmenneen peliyhtiön hieno pelihistoria saa ansaitsemaansa muistelua. *Dune: Ornithopter Assault* -peliä työsti alkujaan Soft Brigade -studio Cryo Interactiven leivissä. Ranskalainen Cryo oli kerännyt mainetta 1990-luvun alkupuolella upeilla PC- ja Amiga/Atari-seikkailuillaan, joista firma siirtyi tyylikkäästi FMV-aikakauden kautta uuden sukupolven PC- ja konsolipelien maailmaan. Firman suurimpiin hitteihin lukeutui huima *Dune*-seikkailu (1992), jonka menestystä firma yritti sittemmin lämmitellä useampaankin otteeseen. Seikkailupeli *Frank Herbert's Dune* (2001) kuitenkin floppasi ja Cryo romahti taloudellisiin ongelmiinsa, minkä myötä myös *Dune Generations* ja *Dune: Ornithopter Assault* kuopattiin.

20 vuotta myöhemmin *Dyynin* sudenkorentokopterit lentävät sittenkin Game Boy Advance -alustalla. Pieni mutta onnistunut joukkorahoitus varmisti ornitopteriräiskinnän valmistumisen nimellä *Elland: The Crystal Wars*. Nyt tosin ei taktikoida tai juonita, vaan planeetan taivailla käydään suoraviivaisempaa sotaa eri faktioiden välillä. Kehittäjä/julkaisija Retro Room Games arvelee saavansa pelin julkaisukuntoon tänä keväänä, ja peli on jo mahdollista ennakkotilata niin kaikki herkut sisältävänä laatikkona, GBA-moduulina tai digitaalisena ROM-painoksena osoitteesta theretroroomgames.com/products/elland-the-crystal-wars.

SUOSIKKIÄLYPELI WORDLEN RETROSAATIO

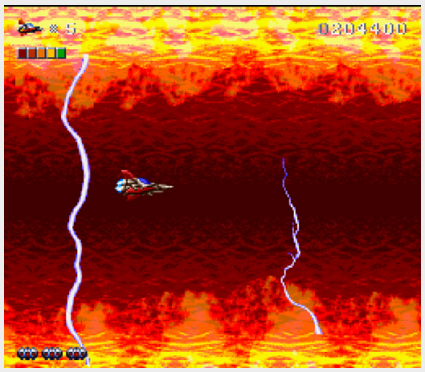
Ilmaisesta selainpelistä maailmanlaajuiseksi somehitiksi kasvanut *Wordle*-sanapeli on taantunut myös vanhemmille pelialustolle. Wordleämään pääsee jopa Game Boylla! Ilmaisessa käännöksessä ei ole muistirajoitteiden takia alkuperäispelin täyttä sanavarastoa, mutta kyllähän 8 000 englanninkielisellä sanalla pääsee ainakin alkuun. Lataa pelin ROM täältä: twitter.com/ghidraninja/status/1490323175510581263.

Myös vauhdikkaat suomisäätäjät tarttuivat Wordleen ja tekivät siitä Commodore 64 -version, *Nepulin*. KozMOS2022-demopartyillä julkaistun pelin taustalla on varsinainen suomiscenen kuka on kukin -tiimi, jolla on vahvaa läsnäoloa myös suomalaisen pelialan timanttisimmassa ytimessä. Kuusnelospelin tekeminen on varmasti ollut rentouttavaa vaihtelua PS5:n sun muiden nykykoneiden pelikoodaamisen lomassa. Lataa peli täältä: csdb.dk/release/?id=214354.

RAINBOW ARTSIN PELIT TULEVAT TAAS

Retrorintamalla rajusti puskeva Ziggurat Interactive tuuttaa koko ajan uutisia uusista vanhoista peleistä. Tällä kertaa yhdysvaltalaiskustantaja on ostanut 1980- ja 1990-luvulla toimineen saksalaisen Rainbow Arts -julkaisijan kuolinpesän, ja omistaa nyt oikeudet yli 80 Rainbowin peliin.

Rainbow Arts ei ollut ehkä maailman saati Euroopan tunnetuin pelitalo, mutta firman värikäs sateenkaarilogo tuli taatusti tutuksi myös suomalaispelaajille. Olihan se kuitenkin yksi manner-Euroopan suurimmista pelitaloista, joka haali riveihinsä myös suurin määrin demoskenestä ponnistaneita tekijöitä ja taitajia.



Vuonna 1984 perustetun yhtiön tunnetuimpia C64-, Amiga-, PC- ja konsolipelejä olivat muun muassa *MadTV, Denaris, Katakis, Giana Sisters, R-Type, To Be on Top* ja *Turricanit 1–3*. Ziggurat ei kuitenkaan kerro täsmällistä listaa ostamistaan pelioikeuksista, mutta uuteen kotiin päätyivät ainakin *Logical, Lollypop, M.U.D.S.,*

MadTV ja *X-Out*. Työn alla on ainakin uudelleenjulkaisu SNESin pelistä *Rendering Ranger: R2*. Siitä julkaistiin aikoinaan vain 10 000 kappaleen painos Japanissa, joten retrokonsolikansalla saattaa olla mielenkiintoa harvinaisuuden uuteen länsijulkaisuun.

Ja miettikääs miten upeaa olisi, jos Ziggurat päättäisi haastaa Nintendon *Gianalla*! Lisätietoja Zigguratista osoitteessa ziggurat.games.

Playstation 1:lle ja 2:lle julkaistiin aikanaan kymmenittäin roolipeliklassikoita, joiden joukossa oli monia mechateoksia. Esittelemme kolme mecha-JRPG:tä, joista jokaisella on oma lähestymistapansa aiheeseen.

VARHAISTEN PLEIKKARIEN

MECHA-ROOLIPELIT

Teksti: Miika Auvinen

Teknotrilleri, hupsuttelu ja jumalantappopeli – mecha-roolipeleistä on moneksi. Esittelyssä olevat mechapelit ovat kaikki erinomaisia, mutta niiden vahvuudet poikkeavat toisistaan. *Steambot Chronicles* luo hyvää mieltä, kun taas Xenogearsin suola on sen eeppinen juoni. Pelit ovat pitkiä, rönsyileviä ja syviä. Vuonna 2020 länsimaissa julkaistu *13 Sentinels: Aegis Rim* osoittaa, että lajityypissä on tehoa näinäkin päivinä – peli on jo nyt nostettu osaksi Vanillawaren kehittämien kulttiteosten joukkoa.

Front Mission -sarja on eräs lajityypin suunnannäyttäjistä. Retro-pleikkareille on julkaistu useampi *Front Mission* -peli, mutta helpoiten saatavilla oleva ja halvin vaihtoehto on *Front Mission 3*. Peli ilmestyi alun perin vuonna 1999 PlayStationille, mutta nykyään se löytyy PSN Storesta ja on pelattavissa PlayStation 3:lla.

Tiukka mechatrilleri

Front Mission 3 on strategia- ja roolipelin yhdistelmä: pelattavuus vaihtelee valikkoruutujen selailun ja vuoropohjaisen taistelun välillä.

Valikkoruuduissa pelaaja jutustelee pelihahmojen kanssa, selaa nettiä ja varustaa omaa mechaansa. Keskusteluissa näkyvät kauniisti piirretyt taustat ja hahmon animetyylillä piirretty pää, ja tekstilaatikot sen kuin vilisevät. Vuoropohjaiset kamppailut ovat strategiahenkisiä, eli ensin liikutaan pelialueella ruudukon näyttäessä alueet, joille pelaaja ehtii vuoron aikana, ja sitten valitaan ase, toiminta ja tuhottava kohde.

Peli alkaa Kaakkois- ja Itä-Aasiasta. Kaksi valtakuntaa, Oceania Cooperative Union (OCU) ja the People's Republic of Da Han Zhong (DHZ), käyvät eräänlaista kylmää sotaa. Filippiineillä, Indonesiassa ja Singaporessa on omat itsenäisyysliikkeensä. Vuonna 2112 Japanin puolustusvoimien tukikohdassa tapahtuu räjähdys. Alkaa sotilasteknologian täyteinen trilleri, jonka aikana päähenkilöt Kazuki Takemura ja Ryogo Kusama kiertävät maailman eri konflikteja.

Pelaajalle on tarjolla kaksi toisistaan poikkeavaa juonilinjaa, uniikkeine tapahtumineen ja dialogeineen. Pelin maailmaa avataan esimerkiksi omalla internetillä. Sähköposteja, valtakuntien eri virastojen kotisivuja, viihdesivuja ja muita verkkokohteita lukemalla pelaaja voi tutustua eri tahoihin ja hahmoihin, joita ei pelissä muuten aina avata.

Kevyttä taktikointia

Mechaansa voi kohentaa aseilla ja osilla. Jokainen osa on itsenäinen palasensa, jolla on omat ominaisuutensa. Vas-

tustajan mechaa ei ole pakko tuhota kokonaan, vaan tämän voi pakottaa antautumaan rampauttamalla sotakoneen. Joskus vihollispilotti hyppää ulos mechastaan, jolloin pelaaja voi vallata aution jättirobon. Eri vihollisilla on erilaiset taistelutyylinsä ja strategiansa: joku taistelee viimeiseen mieheen täydellä aggressiolla, toinen on varautuneempi ja hyökkää vain kun on pakko.

Pelissä pärjätäkseen ei tarvitse olla strateginen nero. Monesti maksimaalinen tulivoima on se juttu, jolla pärjää. Tarinassa ei ole liikoja rönsyjä, ja juoni puksuttelee eteenpäin ilman tarvetta tasojen nosteluun tai muuhun, vaikka mukana onkin kaksi erään alkupään valinnan mukaan valikoituvaa juonilinjaa.

Front Mission 3 on pätevä taktinen roolipeli. Se näyttää ja kuulostaa hyvältä, ja sen juoni sijoittuu virkistävästi meidän maailmamme tulevaisuuteen. Jos tekstilaatikot ja ilmeilevät animenaamat ovat oma juttu, ja vuoropohjainen kevyt strategiakaan ei tylsistytä, peliä kannattaa kokeilla.

Tapetaan Jumala

Vuoden 1998 scififantasiateos *Xenogears* on yksi rakastetuimmista japanilaisista roolipeleistä. Peliä ei julkaistu Euroopassa, mutta se käännettiin englanniksi Yhdysvaltojen markkinoita varten.

Perinteinen ylhäältä kuvattu maailmantutkimiskuvakulma on tuttu monista muistakin Japani-roolipeleistä. Kaupungeissa tutkitaan huoneistoja arkkujen varalta ja jutellaan ihmisille. Maailmankartalla voi liikkua jalan tai kulkuvälineellä. Vuoropohjaiset taistelut kamppaillaan joko nyrkein tai mechan puikoissa. Kamppailusysteemi on yksi JRPG-maailman hienoimmista: eri napit edustavat eri iskuja, ja pelaaja voi myös luoda kokonaan oman kombonsa. Tietyt yhdistelmät tuovat esille erikoisiskuja.

Peli on osa Squaresoftin 90-luvun värisuoraa, jonka ässiin voisi lukea pelejä kuten *Chrono Trigger* ja *Final Fantasy VII*. Monilla sivustoilla kerrotaankin, että *Xenogearsin* oli alun perin tarkoitus olla Final Fantasy- ja myöhemmin *Chrono*-sarjan jatko-osa.

Vaikka kyseessä on retro-Pleikkarin mecharope, taistelut ovat pelissä vain mauste. Pelin hienous on sen tavassa yhdistellä scifiä ja fantasiaa korkealentoiseen filosofiseen sekä uskonnolliseen tematiikkaan.

Se juoni ja se musiikki!

Teoksen juoni käsittää tapahtumia 10 000 vuoden ajanjaksolta. Tarina rönsyilee, ja siinä pohditaan jumaluuksia ja jälleensyntymän kaltaisia uskonnollisia ilmiöitä. Pelkkä pelin pelaaminen ei kerro koko juonta, ja pelin hardcorefanit lukevat myös Perfect Works -kirjasen, jossa sukelletaan syvälle pelin juoneen ja maailmaan. Kirjanen maksaa Ebayssa noin 100 euroa.

Alku on täynnä roolipelikliseitä: tämäkin peli alkaa muistinsa menettäneen nuoren miehen edesottamuksilla, ja hänen kotikylänsä Lahan tietysti tuhotaan muun sivussa. Amnesiasta kärsivä Fei Fong Wong kärsii flashbackeista ja aiemmat elämät riivaavat miestä.

Xenogearsin juoni lainailee ajatuksia länsimaisilta ajattelijoilta, esimerkiksi **Carl Gustaf Jungilta, Sigmund Freudilta** ja **Friedrich Nietzscheltä**. Juoni jakaa mielipiteitä: toisista uskonnollinen, filosofinen ja psykologinen asetelma on videopelijuoneksi syvällinen. Japanilaisten roolipelien faniryhmissä peliä syytetään myös lukiolaistason filosofiasta ja tooran ryöstöviljelystä.

Xenogearsin soundtrack on ikimuistoinen. Säveltäjä **Yasunori Mitsuda** loisti jo vuonna 1995 *Chrono Trigger* -videopelin soundtrackillään. *Xenogearsissa* hän nokittaa jopa aiemman mestariteoksensa vivahteikkaammilla ja draamallisimmilla sävelillä.

Pelin musiikkia kuunnellessa olo on mystinen, ja sävelten tahdissa voisi hyvin lähteä tuhoamaan korkeampia voimia. Mukana on vaikutteita etnomusiikista ja uskonnollisesta musiikista. Kuuntelen usein näiden biisien uudelleenversiointeja, joita etsivä löytää suoratoistopalveluista.

Suunnannäyttäjä

Xenogearsin tekijät ovat luoneet uusia Xeno-alkuisia pelisarjoja, *Xenosagan* ja *Xenoblade Chroniclesin*. Teemat, juonikuviot sekä hahmosuunnittelut tuntuvat toistuvan uudemmissakin sarjoissa – uusissa osissa on myös mechoja. Xeno-sarjan peleissä on paljon silmäniskuja aiemmille osille.

Yksi syy, miksi Xeno-pelien yhteydet eivät ole virallisia on se, että jokaisella pelisarjalla on oma julkaisijansa. Lisäksi konsoli vaihtuu lennossa: *Xenogears* ja *Xenosaga* -pelit julkaistiin PlayStation-konsoleille, kun taas *Xenoblade*-sarja on Nintendo-julkaisu.

Xeno-pelien isänä pidetään **Tetsuya Takahashia**. Hän on kirjoittanut, tuottanut ja ohjannut Xeno-pelejä, ja hän johtaa *Xenosaga* ja *Xenoblade* -sarjat luonutta Monolith Soft -yritystä.

Xenosaga on elämää suurempaa ja lennokasta avaruusoopperaa. Sarja on kolmeosainen, ja osat ilmestyivät PlayStation 2:lle vuosina 2002, 2004 ja 2006. *Xenoblade Chronicles* -sarjassa saippuaooppera siirtyy avoimeen maailmaan. Pelit ilmestyivät Wiille, WiiU:lle Nintendo 3DS:lle ja Nintendo Switchille. Uusin versio sarjasta on vuoden 2020 uudelleenversiointi *Xenoblade Chronicles: Definitive Edition*.

Xeno-pelit ovat olleet monelle pelaajalle ensikosketus syvempiin filo-

sofisiin tai uskonnollisiin näkökulmiin. Eräälläkin fanisivulla eräs sarjan ystävä kertoo, että ennen syvän päädyn scifikirjallisuuden tai taide-elokuvien kokemista, *Xenogears* mullisti kaiken ja jäi mieleen pyörimään päiviksi. Sarjaa voisi verrata *Matrix*-elokuviin: noviiseille filosofiset näkökulmat ovat tuoreita, mutta enemmän filosofiaan tutustuneet ovat jo nähneet vastaavaa pohdiskelua aiemminkin.

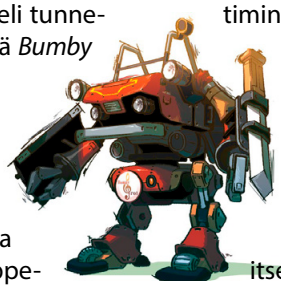
Steampunk-hupsuttelua

Steambot Chronicles on aliarvostettu mecharoolipeli. Teoksessa kierrellään avointa pelimaailmaa popbändin mukana, taistellaan mechataisteluita ja istutaan romanttisilla treffeillä – voiko muuta toivoa?

Peli ei ole järin tunnettu, sillä se ilmestyi Euroopassa vuonna 2006, uuden konsolisukupolven kynnyksellä, aikana, kun JRPG ei muutenkaan ollut se kuumin lajityyppi. Peliin perehtyneet usein kehuvat sen omalaatuisuutta. Kyseessä ei ole jungilaisella psykologialla leikittelevä jumalantapposeikkailu, vaan hieman hupsu hyvän mielen videopeli.

Steambot Chroniclesin alaotsikko alkuvalikossa on "Romantic Scrap Metal Adventure", romanttinen romumetalliseikkailu. Peli alussa päähenkilö Vanilla on JRPG-kliseiden mukaisesti menettänyt muistinsa. Hän löytää hylätyn höyrymechan ja siinä sivussa törmää naispääosaan, Corianderiin. Pelin avainhahmojen nimet viittaavat kasveihin, yrtteihin tai mausteisiin.

Peli on Irem-yhtiön käsialaa ja sen julkaisi Euroopassa 505 Games. Yhdysvalloissa pelin julkaisi Atlus, japanilaisen roolipelaamisen jumaljulkaisija, ja tämän vuoksi moni lukee *Steambot Chroniclesin* yhdeksi Atluksen kokeellisista hassutteluista. Peli tunnetaan Japanissa nimellä *Bumby Trot*.



Kolmen konstin pelimanni

Pelissä on käytännössä kolme erilaista mekaniikkaa. Simppeleintä toiminta on jalan kulkiessa. Tällöin Vanilla juoksee siihen suuntaan minne tatti osoittaa, yhdestä napista jutellaan ihmisille ja toisesta mennään valikkoon syömään tai vaihtamaan vaatteita. Arkuista ja kaupoista saa jos jonkinmoista kudetta ylleen, ja asu vaikuttaa hahmon maineeseen.

Mechaa ohjataan kahdella tatilla siten, että eteenpäin mentäessä työnnetään molempia tatteja eteenpäin, ja mecha kääntyy jos käyttää vain toista tattia. Pelimaailma on laaja, ja alueelta toiselle liikkuminen tapahtuu mechalla. Mechaosuudet karkoittavat kaavoihin kangistuneita pelaajia kontrollien erikoisuuden vuoksi.

Myös toimintaosuudet, joissa hakataan miekoilla ja ammutaan erilaisilla aseilla, tapahtuvat mechaa käyttäen. Mukana on mechaturnaus ja laajojen erämaiden steampunk-roistot hyökkäilevät kimppuun.

Kolmas mekaniikka on musisointiminipeli, ja yksi pelin yksi tärkeimmistä alavirtauksista onkin The Garland GlobeTrotters -bändissä soittaminen. Vanilla voi treenata eri soittimia huuliharpusta haitariin. Pelissä on kuusi musisoitavaa kappaletta. Moni on kehunut pelin musiikkia, mutta itseäni *Steambot Chroniclesin* juustoiset rakkauslaulut lähinnä huvittavat. Toisaalta jämäballadit sopivat pelin muuhun tunnelmaan erinomaisesti, eikä teosta tule ottaa liian vakavasti.

Valinnoilla on väliä

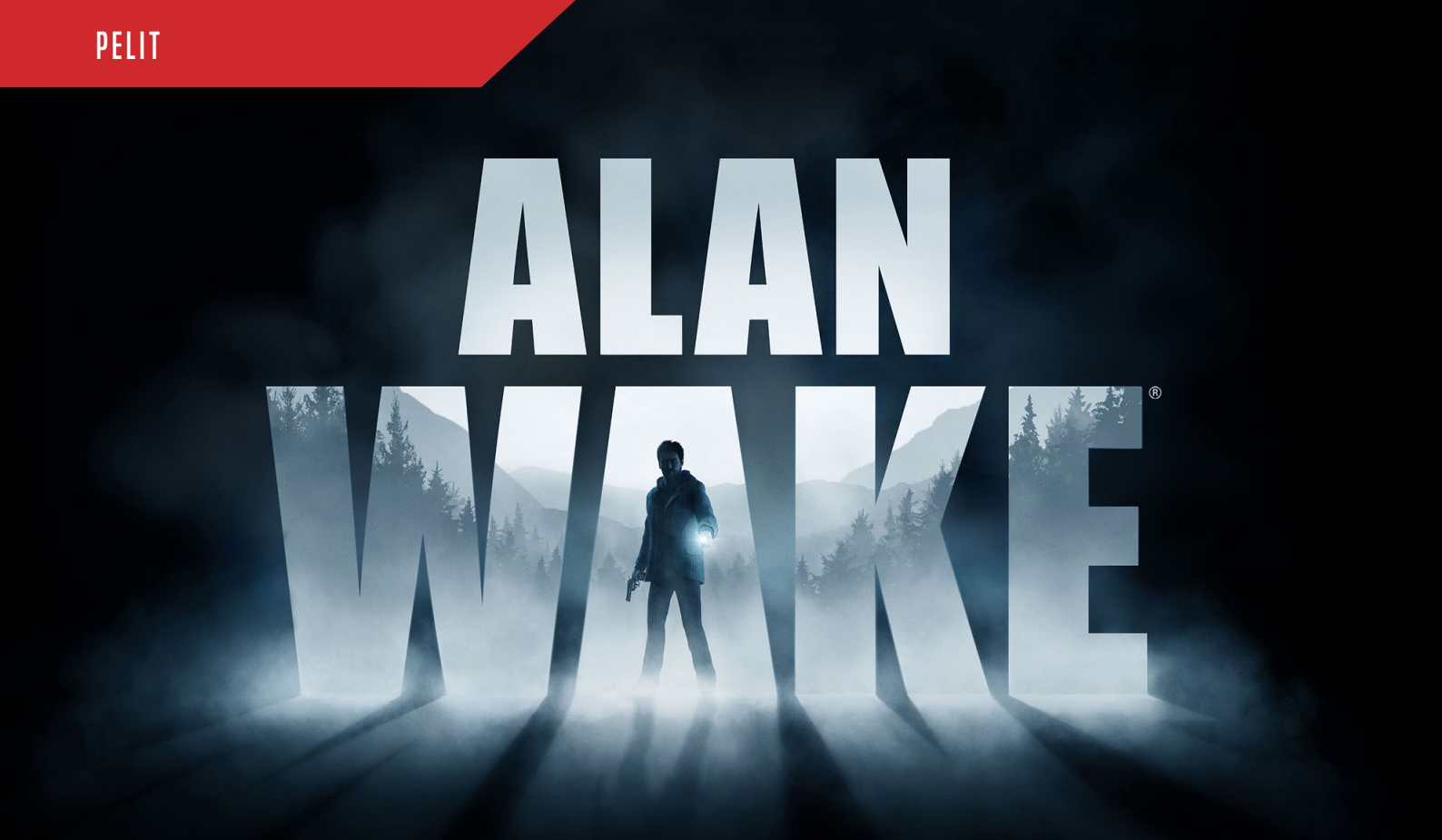
Yksi *Steambot Chroniclesin* vahvuuksista on pelaajalle tarjotut vaihtoehdot: kepeimmillään pelaaja saa valita, haluaako tämä muita hahmoja tervehtiessään nyökätä vai suudella heidän kättään. Syvimmillään valinnat vaikuttavat pelin suuriin juonenkäänteisiin.

Mecharymistely voi olla pelaajan valinnoista riippuen hyvän mielen tarina tai astetta synkeämpi tapahtumaketju. Pelaaja voi valita kuinka kohtelee pelin naishahmoja treffeillä, ja välillä pohditaan, liittyäkö terroristijärjestöön. Lopputulema voi olla synkeäkin. Eräässä lopussa The Garland GlobeTrotters -bändi hajoaa ja yksi sen jäsenistä päätyy asumaan sillan alle.

Pelaaja voi tehdä myös esteettisiä valintoja ja omaa mechaansa voi muokata monin tavoin. Skorpionijalat, musta väri ja miekat molemmissa käsissä luovat aivan erilaista tunnelmaa kuin pelin alussa, jolloin keltainen romumecha löytyy.

Paketti ei ole täydellinen: peli on nykyään hintava käytettyjen pelien markkinoilla ja hupsuttelu ei varmasti iske vakavamman JRPG-jatkumon ystäville. Voin kuitenkin suositella sitä vilpittömästi heille, joista juustoisia balladeja sisältävät bändiosiot ovat juuri parasta, ja heille, joista on kiva koetella pelin rajoja tekemällä välillä luikurimaisia valintoja.

Vuonna 2008 ilmestyi *Steambot Chronicles: Battle Tournament* PSP:lle. Varsinaista jatko-osaa *Steambot Chronicles* ei kuitenkaan koskaan saanut.



REMASTERED

KUUMEHOUREINEN MATKA SUOMALAISEN KAUHUTRILLERIN YTIMEEN

Teksti: Jukka O. Kauppinen Kuvat: Jukka O. Kauppinen, Remedy Entertainment

Minä ja Alan Wake olemme olleet aina kavereita. Yhteinen taipaleemme on kuitenkin ollut myrskyinen, jopa painajaismainen.

Myös suomalaisen Remedy Entertainmentin taival on ollut värikäs. Äkkiseltään voisi sanoa sen olevan hittien täyteinen, mutta onhan kulissien takana tapahtunut yhtä jos toista. Työn alla on ollut monta muutakin peliä kuin ne lopulta julkaistut. Pelibisnes on showbisnestä – aina tapahtuu ja vain harvasta tulee tähti, hitti.

Remedy julkisti jokin aika sitten tekevänsä jatko-osaa muinaiselle klassikolleen, *Alan Wakelle.* Se oli hieno uutinen Remedyn ystäville, sillä Alanista on kasvanut yksi studion rakastetuimmista peleistä. Samalla *Alan Waken* taival on kuitenkin ollut kuoppainen. Pelin julkaissut Microsoft halusi suomiseikkailusta yksinoikeusvalttikortin Xbox 360 -konsolille ja Windowsille – mutta vain ja ainoastaan kurjan alun saaneelle Windows Vista -käyttöjärjestelmälleen. Vista-pakotus yökötti jo etukäteen osaa PC-pelaajista, mutta sitten käännös kuopattiin. Liekö Microsoft pelännyt pelaajien Vista-vihaa vai mitä?

Xbox 360 -versio kuitenkin ilmestyi ja oli huikea peli, joka myi ihan OK mutta hissukseen. Jättihitti jäi saamatta, vaikka peliin pureutuneet löysivätkin pinnan alta koukuttavan seikkailun. *Alan* jäi kulttihitiksi, kunnes Remedy sai lopulta pelin oikeudet takaisin Microsoftilta ja teki siitä PC-version omin nokkinensa, ilman Only for Vista -pakkopullaa. Ja oi, sitä pelaajakansa olikin odottanut. *Alan Waken* PC-versio tienasi itsensä takaisin vuorokaudessa ja sen myynnit ohittivat railakkaasti Xbox-myynnit. Miniatyyrijatko-osa *Alan Wake's American Nightmarekin* myi kohtuullisesti. Kaiken kaikkiaan *Alan*-sarjan myynnit ovat nousseet ajan kanssa tyydyttäviksi, mutta kyllä siinä kestikin.

Alanin matkassa

Minulle *Alan Wake* on ollut julkaisustaan lähtien lähes täydellinen peli. Se oli tiivis ja upeasti käsikirjoitettu elämys, joka hivelsi minua juuri oikeista paikoista. Olen aina nauttinut hyvistä seikkailuista ja ennen kaikkea tarinoista. *Alanissa* on molempia, roppakaupalla, harvinaisella tavalla yhteen pakattuna - ja miellyttävillä, kiinnostavilla pää- ja sivuhenkilöillä höystettynä.

Alan Wake on tarina tavallisista ihmisistä - he eivät ole supersankareita, eivät asehulluja tai masentavia ihmishirviöitä.

Kuten **Sami Järvi** minulle aikoinaan ynnäili:

- Peli on hyvin henkilövetoinen ja siinä on voimakas päähenkilö. Halusimme sankariksi jokamiehen, ei ammattimaista toimintasankaria vaan hahmon, jonka täytyy kasvaa rooliin.

Alan Wake oli moniongelmainen ja vaikea tyyppi, joka onnistui sössimään elämänsä jo ennen kuin saapui Bright Fallsiin. Kun joku on äkkipikainen ja itsekäs, se tuottaa helposti kiinnostavia jännitteitä. Lopulta hahmon on kuitenkin pakko katsoa peiliin. Kirjoittajalle kusipäisyys on yleensä kiinnostavampaa kuin kirkasotsaisuus.
*– **Mikko Rautalahti***

Pelin maailma on tavallinen maailma. Ehkä eräällä tavalla vähän sonkajärvimäinen - sivussa, syrjässä ja ylpeä omista pienistä ihmeellisistä perinteistään. Bright Fallsissa on ehkä vähän enemmän nähtävää, mutta eiköhän kuppilakahvi ole yhtä pahaa maailman joka kolkassa.

Kaikessa kauniissa on silti varjopuolensa. Bright Falls on myös Darkness Falls. On päivä, on yö. Paikalla on salaisuutensa ja kaiken takana vaanii usva, jonka synkkyys on valmis nielemään kitaansa myös kiltit, kunnon ihmiset. He eivät pelaa kaksinaamaista peliä, vaan raukkojen sielu on nielaistu – tilalla on vain marionettejaan liikuttava pimeyden voima, joka koettaa työntää Alania synkkyyden ytimen haluamaan suuntaan. Todellisuus ja taru sekoittuvat.

Alan Wake on jännäri. Päätimme tietoisesti tehdä hyvän trillerin eikä kauhupeliä. Veri, suolenpätkät ja hirviöt eivät ole pääosassa, vaan mysteeri, tunnelman rakentaminen ja tuntemattoman pelko.
- Sami Järvi

Yhä kekseliäs

Alan Waken suurin voima ja innovaatio ei ole bullet time tai mikään muukaan peliteknologinen keksintö, vaan kahden vastakkaisen voiman valjastaminen voimaksi ja vaaraksi – toivoksi ja peloksi. Valo vastaan pimeys. Bright Fallsin pimeys – Dark Presence – saattaa hyökätä missä ja miten tahansa, jonka jälkeen pelaaja tietää täsmälleen missä mennään ja mitä on luvassa. Mitä pimeämpää, sen pahempaa. Mutta jokainen valonpilkahdus pimeässä on toivon majakka. Siellä on turvapaikka, ensiapua ja ehkä jotain kättä pidempää.

Valo on myös ase, kirjaimellisesti. Pimeys suojaa valtaamiaan hahmoja, eivätkä tavalliset aseet pure heihin. Valo, erityisesti voimakas valo, kuitenkin saa pimeyden hahmot kavahtamaan ja heikentymään niin, että nämä tottelevat kättä pidempää.

Alanin seurassa matkaaminen onkin aivan omanlaisensa kokemus, etenkin kun pelissä on kaksi niin täysin erilaista puolta. Päiväsaikaan asiat ovat lähestulkoon normaalisti. Pelaaja kulkee tutkimassa paikkoja, kerää lisätietoja Bright Fallsin mysteereistä ja kohtaa hyvin, hyvin mielenkiintoisia ihmisiä. Yöllä tai pimeyden hyökätessä hän on uhri, joka lähinnä pakenee ja yrittää selviytyä hengissä. Pelaajan ympärille rakennetaan tiiviin jännityksen, säikähdysten ja hyytävän odotuksen kudelma, joka on lopulta aika pelottava, hyvällä tavalla. Seikkailu ei silti pohjaudu pelotteluun tai kulman takaa tyhjästä ilmestyvien hirviöiden loputtomaan lahtaamiseen, vaan elokuvamaisesti rakennettu käsikirjoitus johdattelee aina kohti seuraavaa toivonpilkahdusta.

Silti on myönnettävä, että parhaimmillaan tai pahimmillaan oloni Alanin matkassa on ollut jopa raskasta. Etenkin kerran sairastuessani, kun tempaisin jostain mukaani todella rajun kuumetaudin. Olo oli täysin veltto ja voimaton, mitään ei jaksanut tehdä. Pelituolissa rötköttäminen sentään onnistui, joten pelasin houreissani *Alan Wakea* Xbox 360:llä. Siitä seurasi jotain aika omituista. Pelin ahdistavat pahuuden usvapilvet seurasivat minua myös pelin ulkopuolelle. Yritin nukkua, mutta näin kuinka pilvet kiemurtelivat kattolampun ympärillä ja pursusivat seinävaatteen takaa kiertelemään makuuhuoneessa. Olin hereillä. Tiesin, että pilvet eivät olla oikeita. Ne katosivat, kun sytytin valot – ja palasivat, kun sammutin ne. Makasin sängyssä kuumehikilammikossa, tuijotin Darkin usvien kiertelyä ja huusin ääneen: perkel**n Remedy! Haistakaa p*ska*, Remedy!

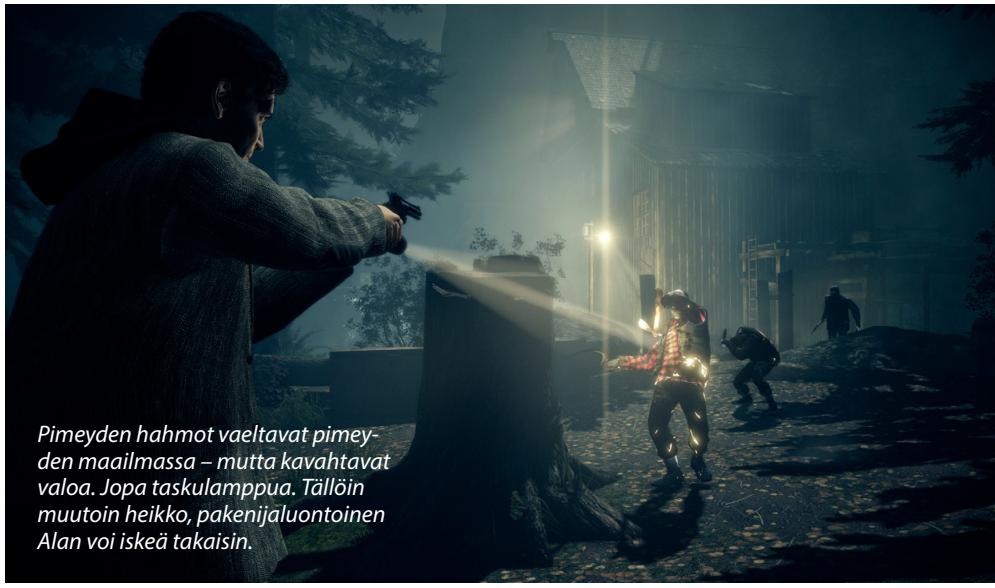
En nukkunut sinä yönä, enkä jatkanut *Alan Waken* läpipeluuta ennen kuin olin taas terve. Kerroin remedyläisille elämyksestäni. He arvostivat.

Taitavaa designia

Vuonna 2022, *Remasterin* läpipeluun aikana, opin arvostamaan *Alanin* pohjalla olevaa taitavaa pelisuunnittelua uudella tavalla. Se osoittautui edelleen toimivaksi ja nautinnolliseksi peliksi, jossa on lumoava rytmitys, niin tarinan kuin toiminnan suhteen. Peli tarjoaa kiireettömiä hetkiä, joiden aikana voi tutustua Bright Fallsin asukkaisiin ja Alaniin henkilönä, ihmisenä, samoin kuin kaupunkiin ja pelaajalle rakennettuun ympäristöön. Välillä matkaa taitetaan jalan tai autolla, maisemista nautiskellen. Myös pimeyden keskellä matkaamiseen on varattu runsain mitoin aikaa – aikaa pelätä ja olla huolissaan. Joskus mitään ei tapahdu. Pelaajaa vain piinataan, mutta kun toiminta viimein alkaa: silloin vasta puhutaankin rytmityksestä. Kulkemista, pakenemista, yhteenottoja, taisteluja, sydämentykytystä ja juuri kun tuntuu, ettei pumppu enää kestä – pelastava valo, turvasatama, tauko.

Erilaisista ideoista meillä ei ollut pulaa, kyse oli enemmänkin siitä, mitkä sopivat kokonaisuuteen, ja mitä ylipäätänsä lopulta ehdittäisiin toteuttaa.
– Mikko Rautalahti

Vaikka *Alan* ei ole kauhupeli, on se äärimmäisen taitava trilleri. Se saa sydämen jyskyttämään, mielen jännittymään ja tunteet kohoamaan. Mutta se ei ole kuolemanvakava, vaikka onkin. Tarinassa on kevennyksensä, joista herkullisinta antia ovat *Twilight Zone* -ohjelmasta ammentava Night Springs -kauhusarja. Aina joskus, jossain, pe-



Pimeyden hahmot vaeltavat pimeyden maailmassa – mutta kavahtavat valoa. Jopa taskulamppua. Tällöin muutoin heikko, pakenijaluontoinen Alan voi iskeä takaisin.

Saisiko olla kahvia? Syvää ja mustaa, ilman maitoa. Todella Darkiksi paahdettuna.

laaja kohtaa television, josta hän voi katsoa muutaman minuutin tv-jakson. Kauhu-telkkarisarjaa tv-sarjan tapaan rakennetussa kauhupelissä? Yo dawg.

Me halusimme, että pelottavat asiat ovat pelottavia vielä lopussakin
– Sami Järvi

Interaktiivinen elokuva

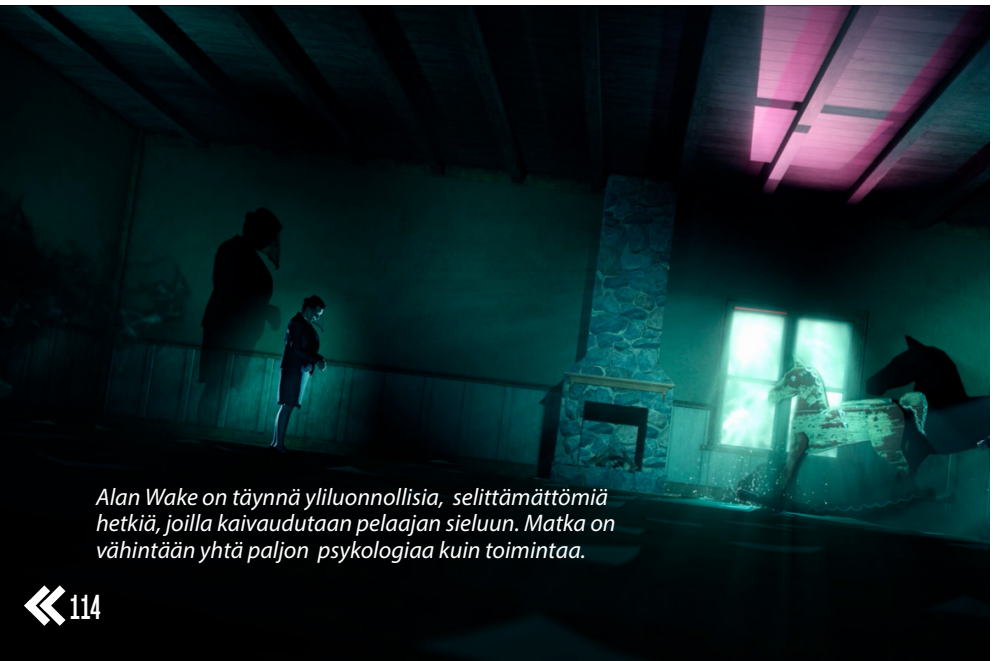
Remedy on pyrkinyt suurissa peleissään tuomaan videopelien maailmaan elokuvallista kerrontaa mutta interaktiivisessa muodossa. Tuoreempi *Quantum Break* oli tästä huikea esimerkki, mutta *Alanissa* käsikirjoittajat olivat mahtavassa kehitysvaiheessa. Alun avoimen maailman seikkailu muuntui pitkäksi venyneiden kehitysvuosien aikana erilaisia teemoja tarjoaviksi tv-sarjamaisiksi jaksoiksi – ja käsikirjoituksellakin oli pituutta kokonaisen tv-sarjan kauden verran. Ei ihme.

- Käsikirjoitusformaatti on hyvin lähellä elokuvaformaattia, mutta pelissä on valinnaista materiaalia. Seuraako pelaaja juonta vai meneekö eteenpäin, se muovaa osaltaan kokemusta, Järvi muisteli.

Alania voikin pelata eri tavoin: pikajuoksijat eivät huomaa maailmaan ja kerrontaan upotettuja sävyjä ja yksityiskohtia, joiden kautta on mahdollista laskeutua syvemmälle pelin syövereihin. Halutessaan tarinaan voi uppoutua kuin syvämerensukeltaja, hartaasti ja ajatuksella. Pimeyden ytimessä fiilistelyä sävytetään kertojalla, joka avaa suurennuslasia Alanin ajatuksiin ja tuntemuksiin. Pelihahmon oman äänen käyttö kertojana onkin kasvanut yhdeksi Remedyn tavaramerkeistä.

- Näin jälkikäteen miettien se, että Alan Waken oli alun perin tarkoitus olla avoimen maailman peli, antoi meille tiettyjä tarinankerronnallisia etuja. Koska pelimaailmassa riitti lääniä, kentät saattoivat olla mittakaavaltaan hyvin laajoja. Se antoi mahdollisuuksia tarinalliseen fiilistelyyn ja rytmitykseen: kohtausten rakentamiseen löytyi yleensä mukavasti tilaa, ja niitä oli mahdollista pedata pitkään. Se on luksusta, jota ei usein tämän päivän AAA-peleistä löydy, kun jokaisen rakennetun neliömetrin hintalappu on niin kova. Nykyisin on vaikea argumentoida, että nyt pitäisi saada tähän väliin minuuttikaupalla pelkkää metsässä fiilistelyä.
– Mikko Rautalahti

Alan Wake ei ole pelkkä peli. Se on suomalaisen peliteollisuuden mestariteos ja klassikko, joka on tänään muutakin kuin kulttiklassikko. Se on rakastettu ydin, jonka ympärille Remedy rakentaa omaa, pelistä toiseen kantavaa maailmaansa. Ja vaikka *Alanin* uusi *Remastered*-versio ei ole yhtä helmeilevän kaunis kuin monet nykypelit, se on yhä upean koukuttava seikkailu omaperäisellä toiminnalla ja sykähdyttävillä, viehättävillä sivuhenkilöillä varustettuna. Se on yhä peli, jonka läpipeluusta nauttii.



Alan Wake on täynnä yliluonnollisia, selittämättömiä hetkiä, joilla kaivaudutaan pelaajan sieluun. Matka on vähintään yhtä paljon psykologiaa kuin toimintaa.

Alan Wake
Kehittäjä: Remedy Entertainment
Julkaisija: Microsoft Game Studios
Vuosi: 2010
Alustat: Xbox 360

Alan Wake's American Nightmare
Kehittäjä: Remedy Entertainment
Julkaisija: Microsoft Game Studios
Vuosi: 2012
Alustat: PC, Xbox 360

Alan Wake
Kehittäjät: Nitro Games, Remedy
Julkaisija: Remedy Entertainment
Vuosi: 2012
Alustat: PC

Alan Wake Remastered
Kehittäjä: D3T & Remedy
Julkaisija: Epic Games
Vuosi: 2021
Alustat: PC, PS4, PS5, X1, XSX|S

Tulossa: Alan Wake 2
Milloin: When it is done.

B-17 FLYING FORTRESS: WORLD WAR II BOMBERS IN ACTION
B-17 FLYING FORTRESS: THE MIGHTY EIGHT

MicroProsen muinaiset *B-17*-simulaattorit olivat outolintuja. Ne eivät nimittäin olleet lentosimulaattoreita, vaikka yhtiö olikin rakentanut maineensa simulaattoriputiikkina. Nykytermein kuvattuna ne olivat enemmänkin resurssien manageerointi -strategiapelejä, joissa pelaaja yrittää auttaa lentokoneen miehistöä toimimaan parhaalla mahdollisella tavalla toisen maailmansodan taivailla.

Helppoahan se ei tietenkään ole – mikä olisikaan kolmannen valtakunnan taivailla teräslinnun kyydissä liidellessä, kranaattien raadellessa konetta, liekkien nuollessa siipiä ja miehistön vuotaessa kuiviin. Onneksi veri ei kerry koneen pohjalle liukkaiksi lammikoiksi, vaan valuu ulos luodinrei'istä.

B-17-simulaattorit on julkaistu alkuperäisessä muodossaan uudelleen netin PC-pelipalveluihin. Sisältö ja tekniikka ovat entiset, pelit on vain puukotettu toimimaan nykykoneilla. Jälki ei ole kovin hienovaraista, eivätkä muinaispelit tarjoa moderneja hienouksia. Silti jälki miellyttää, ainakin paikoin. Kaltaiseni simulaattori- ja strategiafani löytää molemmista *B-17*-strategiasimuista hyviä puolia ja kelpo pelattavaa, joskin täytyy tunnustaa, että vanhempi *Bombers in Action* viehättää enemmän hellyttävällä kömpelyydellään.

Yksinkertaisimmillaan sotalento Englannista Saksan päälle on myhäilyä ja päällekatsomista. Pommikoneen miehistö suorittaa tehtäviään automaattisesti, ilman ihmispelaajan päsmäröintiä. Pelaaja voi halutessaan siirtyä myös kenen tahansa miehistön jäsenen tilalle: vaikka lentämään konetta tai ampumaan hyökkäävät hunnit taivaalta konekivääreillä. Tilanne muuttuu kuumemmaksi, jos koneessa syttyy tulipalo – joku pitää nopeasti saada sammuttamaan sitä jauhesammuttimella. Kuka ehtisi antamaan ensiapua haavoittuneelle radistille? Apua, lentäjä sai konetykistä kalloonsa – pystyykö joku lentämään koneen takaisin Englantiin ja jopa laskeutumaan sillä?

Nätti VGA-aikakauden pikseligrafiikka toimii hienosti lentokoneiden sisällä, vähemmän kk-tornissa roiskiessa. Suurin kynnys on henkinen: tämän ajan peleissä ei ole sisäisiä tutoriaaleja tai apuja ja digitaalisesti jaettu ohjekirja on pakko lukea, jos haluaa saada mitään aikaan. Digijakelussa ei tule edes näppäimistön päälle asetettavaa näppäinkarttapahvia. Ei ole helppoa palata nykypäivän laiskajaakkomaailmasta ysäridesignin maailmaan. Riittävän nysväämisen ja ärräpäästelyn jälkeen se on kuitenkin lysti ja palkitsevakin kokemus.

Samaan ei yllä uudempi *B-17*. Se oli jo aito Windows-sovellus, joka tehtiin 3D:n ja multimedian aikakaudella. Onhan siinä hyvääkin: 3D-grafiikat näyttävät lähes kelvollisilta, ja vihollisoneet tunnistaa lentokoneiksi. Hiirtä osataan käyttää jo aika fiksusti käyttöliittymässä ja hauskana sisällöllisenä extrana pelaaja voi lentää myös B-24 Liberator -pommittajalla. Voipa hän istahtaa myös saattohävittäjän tai Luftwaffen torjuntahävittäjien konttoriin. Eihän hävittäjäsimuilu kovin syvällistä ole, mutta kiva sitä on kokeilla.

Ongelmia vain on enemmän. Huonosti digitoidut koneiden sisäkuvat oksettavat, pelaaminen on edeltäjäänsä vaikeampaa ja vuoden 2000 pelisuunnittelussa on panostettu vääriin asioihin. Ruma ulkokultaisuus korvasi timanttisen sydämen, eikä sotalentäminen Eurooppaan ole enää likimainkaan yhtä hauskaa. Silti, kivahan tätäkin on ainakin vilkaista peli- ja ilmasotahistoriallisessa mielessä, etenkin kun peli käynnistyy ja pyörii teknisesti edeltäjäänsä nätimmin.

MicroProsen muinaisuuksien uudelleenjulkaisu on toki kulttuuriteko. Australialainen **David Lagettie** on ostanut vuosien ajan hiljalleen maailmalle hajautuneita MicroProse-pelien oikeuksia, ja onnistui lopulta ostamaan myös pelitalon nimen. Näin kohtuullisen iso osa MPS-historiaa on taas yksissä käsissä. Vaikka nykyisellä MicroProsella ei ole paljoa yhteistä esikuvansa kanssa, ei se kuitenkaan ole Atarin kaltainen tyhjiin pumpattu kulissi, jonka modernimpi tuotanto on ollut valtaosin kuravelliä. Siksi tohdin antaa peleille varovaisen kehupeukun, kunhan muistatte, ettei näiden pelaaminen ole helppoa.

B-17 Flying Fortress: The Mighty Eight

B-17 Flying Fortress: World War II Bombers in Action
Kehittäjä: Vektor Grafix
Julkaisija: MicroProse
Vuosi: 2021 (1993)
Alustat: PC (PC, Amiga, Atari ST)

B-17 Flying Fortress: The Mighty Eight
Kehittäjä: Wayward Design
Julkaisija: MicroProse
Vuosi: 2021 (2000)
Alustat: PC (PC)

TEKSTIT: JUKKA O.KAUPPINEN

CARRIER COMMAND 2

Vuonna 1988 julkaistu *Carrier Command* oli yksi aikansa megapeleistä. Ajoitus oli oivallinen: Amiga oli kovassa huudossa maailmalla, kiitos edullisen Amiga 500 -mallin. *Carrierissä* yhdistyi uuden tehomikron tenho ja uudenlainen pelaaminen: onpa sitä kehuttu RTS-genren edelläkävijöiksi.

Olihan siinä potkua. Pelaaja komensi valtavaa lentotukialusta, jolla hän liikkui laajassa saaristossa vallaten saaria ja valjastaen nämä oman sotakoneistonsa palvelukseen, resurssi- ja tuotantolaitoksiksi. Jossain toisessa päässä oli vihollisen tukialus, jolla oli samat suunnitelmat. There can be only one. Välienselvittely sujui strategia- ja toimintapelin yhdistelmänä, sillä pelaaja pystyi sekä komentamaan tukialustaan ja yksiköitään karttapohjaisesti että ottamaan ne suoraan ohjaukseensa. Manta-hävittäjillä lentäminen olikin aikanaan elämys.

Samaan ei ylletä *Carrier Command 2*:ssa, kuten ei oikein ylletty aiheen aiemmissakaan lämmittelyissä. CC2 on sekamelska monia erilaisia komentojärjestelmiä, joita yhden pelaajan on vaikea hallita. Siinä on ison co-op-moninpelin makua, sillä CPU-tukialuksen etsiminen ja saariston valtaaminen on selvästi rakennettu usean ihmispelaajan yhteistyöelämykseksi. Tukialuksen ja sen erilaisten järjestelmien ja aseiden kontrollit on jaettu niin moneen eri työasemaan, ettei niiden välillä käpyttelyä voi kehua järin nautinnolliseksi.

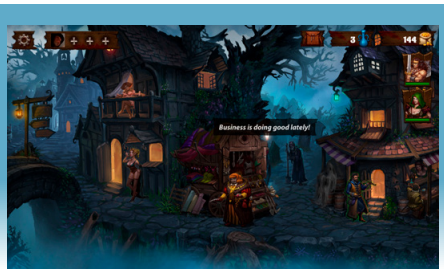
Mukana on toki vaikutteita alkuperäispelistä, peräti parannuksiakin. Pelaaja voi komentaa koneitaan sekä kartalla että hypätä niiden kyytiin tarkkailemaan, jopa ohjaamoon pääsee paukuttamaan. Erilaista tekemistä on enemmän, mutta niin, kaikkea on enemmän, se on hankalampaa ja ennen kaikkea hidasta. *CC2* on turhauttavan hidas – kaikkea joutuu odottamaan loputtomiin, eikä sen enempää strategia kuin toimintakaan ole palkitsevaa. Omat tankit ja lentokoneet ovat lisäksi raivostuttavan tyhmiä, joten jos haluaa kunnon jälkeä, kaikki on pakko tehdä itse.

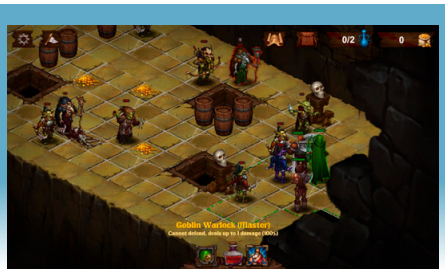
CC2 vaatii pelaajaltaan harvinaisen sitkeää paneutumista ja uskoa lopputulokseen, eikä ainakaan minun pinnani riitä enää siihen. Myös alkuperäinen retroversio on vanhentunut kämäisesti, joten väittäisin *Carrier Commandien* ajan olevan ohi.

CARRIER COMMAND 2
Kehittäjä: Geometa
Julkaisija: Microprose
Vuosi: 2021
Alusta: PC

CARRIER COMMAND
Kehittäjä: Realtime Games
Julkaisija: Rainbird
Vuosi: 1988
Alustat: Amiga, Atari ST, PC, C64, muita

DARK QUEST II





Brittiläisen **Stephen Bakerin** vuonna 1989 julkaistu lautapeli *HeroQuest* oli yksi aikansa suurista sukupolvikokemuksista. Jo seuraavana vuonna suomeksi julkaistu peli oli yksi sosiaalisemman yhteispelaamisen edelläkävijöitä, vähän kuin myöhempien aikojen *Carcassonnet*. 2–5 pelaajan lautapeli oli niin iso hitti, että Gremlin Graphics toi luolaromppaamisen myös kotimikroille. Myös digiversio oli jättihitti ja sitä veivattiin tunteella myös Suomessa, etenkin kun seikkailuun ei tarvinnut kaveriporukkaa.

Eri versioineen peli jätti jälkeensä niin maukkaat jälkituntemukset, että siitä on otettu mallia myöhemminkin. Vastaavaa koko kansan peliä ei ole kuitenkaan syntynyt, joskin *Dark Quest*- ja *Dark Quest 2* -romppaukset pääsevät kohtuullisen lähelle esikuvaansa. Idea on sama: sankarit pudottautuvat luolastoon ja könyävät sitä pitkin poikin keräten mahdollisimman paljon aarteita ja taistellen tiensä voitokkaasti luolaston uloskäynnille. Sitten paikataan haavat, viritellään ryhmän varusteita ja yritetään saada pennit riittämään seuraavan maanalaisen reissun varusteluun.

Syviä hienouksia pelissä ei ole. Toiminta on perusnättiä digitaalista lautapelaamista ilman roolipelimäistä hahmonkehitystä, joskin ryhmän kokoonpanolla eri hahmoluokkineen on toki vaikutusta taktiikoihin. Yksinkertainen liike-iske-naksuttelu vaatii myös harkintaa, sillä selviytyminen on usein hyvinkin täpärällä. Yksittäisen luolan selvittää 15–20 minuutissa, mikä on sopiva kesto pikku välipalapelailulle. Käyttöliittymässä tosin olisi vähän viilattavaa. Läpipeluun jälkeen lisää kenttiä löytyy Steam Workshopista, sillä pelin mukana tulee myös kenttäeditori.

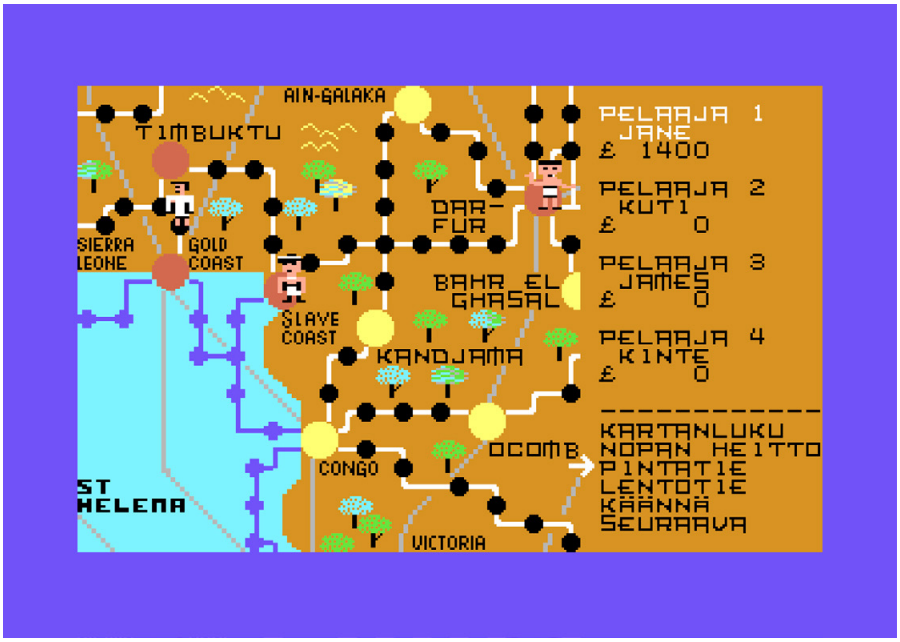
DQ2 sopii hyvin *HeroQuestin* digitaaliseen lämmittelyyn etenkin nyt, kun lämmöllä muistellusta retrolautapelistä ilmestyy tänä vuonna uudistettu painos. Ehkä koko genre saa taas uutta vauhtia!

Kehittäjä ja julkaisija:
Brain Seal
Vuosi: 2018
Alusta: PC, PS4, X1, XSX|S, Switch, iOS, Android

Peli:	AFRIKAN TÄHTI
Vuosi:	1951 / 1985
Alusta:	lautapeli / C64
Tekijä:	Kari Mannerla / Otso Pakarinen ja Jari Heikkinen
Julkaisija:	Peliko / Amersoft

AFRIKAN TÄHTI

Teksti: Jukka O. Kauppinen



Afrikan tähti on kenties tunnetun suomalaispeli, jota on pelattu suomalaisperheissä sukupolvesta toiseen. Se herättää väistämättä hyvinkin monenlaisia tunteita: joillekin se on rakas muisto lapsuudesta, toisille säännöllisesti pelattava perinne ja joillekuille pelkoa ja inhoa aiheuttava kauhistus.

Onhan sitä itsekin tullut pelattua, etenkin lapsena, myöhemmin vain harvakseltaan sopivassa seurassa ironisesti tai lasten kanssa. Silti Afrikan tähti on tärkeä peli. Sen luoja **Kari Mannerla** oli tehnyt pelejä jo 14-vuotiaasta saakka, ja sai idean *Afrikan tähteen* **Humphrey Bogartin** seikkailuelokuvien jälkihuumassa. Mitä jos Afrikassa voitaisiin seikkailla ja etsiä Afrikan Tähti -timanttia? Pelisuunnittelu oli nohevaa: kartalle vain läimittiin jännittäviltä kuulostavia paikannimiä. Ratkaisevia ideoita oli muuttaa kiinteät kulkureitit vapaasti valittaviksi ja tehdä tapahtumanappuloista käännettäviä ja sekoitettavia.

Eihän *Tähti* ole nykysilmin edes hyvä peli. Sen mekaniikka pohjautuu sattumaan ja onnenkantamoisiin. Tähdessä tärkeintä onkin peliseura: kokemus rakentuu yhdessä istumisesta ja jännittämisestä, ei syvällisestä jännittävyydestä, taidoista tai taktiikoista. Ehkä myös paljolti perinteestä. Tätä on ennenkin pelattu, joten pelataan nytkin! Sukupolvelta toiselle! Vaikka väkisin!

Maailma vuonna 1951 oli erilainen. Monet silloisista lautapeleistä olivat aika tyhmiä, jopa rikkinäisiä. Harvan sääntöjä oli luettu edes tekijöiden taholta, saati koepelattu kunnolla. Pelejä puskettiin kauppoihin beta-raakileina. *Tähti* oli sentään ehjä peli, jossa ei ollut tulkinnanvaraisuuksia tai umpikujia, killeribugeja. Ehkä juuri siksi pelistä tuli niin valtava hitti.

Pelimuseossa esillä on *Tähden* pahvisempia versioita, mutta onhan siitä tehty digitaalinenkin painos. Uudemmat Android- ja iOS-versiot näyttävät kadonneen digitaivaaseen, mutta onneksi Amersoftin julkaisema Commodore 64 -painos elää kaikkien harrastajien sydämissä. Nimenomaan sydämissä, sillä ei sekään erityisen nautinnollinen ole nykysilmin ja -joystickein koettuna.

Otso Pakarinen ja **Jari Heikkinen** loivat digitähden vuonna 1985. Aikaansaannosta voi pitää kaikkien koodisankarimitalien arvoisena, sillä kumpikaan ei ollut tehnyt aiemmin ainuttakaan peliä – ja itse asiassa molemmat opettelivat pelintekoon käytetyn *Promal*-ohjelmointikielenkin prosessin aikana. Edes kehitystyökaluja ei ollut entuudestaan, joten sekä peli että sen työkalut luotiin tyhjästä. Tekijät eivät siis olleet mitään turhia poikia.

Suorituksen arvoa nostaa se, että *Afrikan tähden* bittipainos on vielä tänäänkin yllättävän toimiva ja eheä peli. Esimerkiksi tekoälyvastustajat pelaavat järkevästi ja pitävät ihmispelaajan kiireisenä. Mikään äkillisen tai satunnaisen pelihimon tyydyttäjä se ei kuitenkaan ole, sillä etenemistahti on verkkainen. Afrikkaa tutkitaan tassutellen ja ripeästi reagoiviin nykypeleihin tottuneelle komentojen toteutuminen kestää pienen ikuisuuden. Voinee myös sanoa, ettei alkuperäisestä värikkäästä pelilaudasta ole jälkeäkään kuusnelosversion ruskeapalettisessa tulkinnassa. Lautapelin säännöt osaaville peli on selkeä kokonaisuus suomenkielisine käyttöliittymineen.

Afrikan tähdellä on eri versioineen merkittävä asema kotimaisessa pelihistoriassa. Lautapeli on myydyin suomipeli koskaan, C64-versio puolestaan ensimmäinen suomalainen lisenssipeli, samoin kuin yksi ensimmäisistä suomalaisista saati suomenkielisistä kaupallisista videopeleistä. Vaikka jälkimmäinen ei ollutkaan hitti, niin on se sentään tärkeä osa pelikulttuuriamme ja -historiaamme.

Poimintoja pelimuseosta -juttusarjassa esitellään museon makupaloja myös kulissien takaa.

SUOMEN PELIMUSEO, VAPRIIKKI
Alaverstaanraitti 5, Tampere
suomenpelimuseo.com

PUZZLE BOBBLE

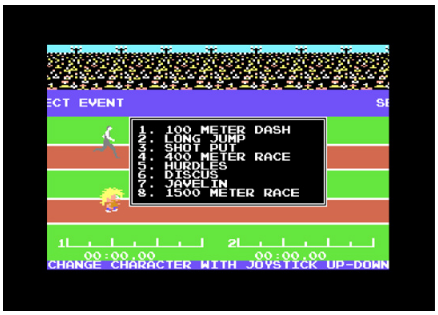
Taiton vuonna 1994 julkaisema *Puzzle Bobble (Bust-A-Move)* on yksi kaikkien aikojen rakastetuimmista ja parhaimmista kaksinpeleistä. Neo Geo -alustalle tehty peli nappasi päähenkilöikseen *Bubble Bobblen* lohikäärmeet ja loi näiden avustuksella herkullisen match-3-puzzletarkkuusammunnan. Ainakin meidän piireissämme se on ollut kestosuosikki, eikä ole ihan sattumaa, että laite löytyy myös Suomen pelimuseon pelihallista.

Puzzlea ei vain ole koskaan nähty virallisena C64-käännöksenä. Ei nytkään, sillä fanit ovat kääntäneet klassikon omin nokkinensa ja lupinensa. Eihän haaste ole teknisesti kummoinen, sillä pohjimmiltaan kyseessä on yksinkertainen pikku peli. Siinä se nerokkuus lymyääkin: yksinkertainen on myös kaunista ja kinkkistä.

Ideahan on laukoa kentälle alareunan tykistä palloja ja luoda niistä vähintään kolmen samanvärisen pallon rykelmiä. Mutta siinäkin on jippoja, kikkoja ja ketjutuksia – ja tarkkaa pelinsisäistä logiikkaa, jonka virheet *Puzzle Bobble* -veteraanit huomaavat. Mutta ei hätää, asiat ovat kohdallaan. Peli toimii kuten pitääkin, lähes identtisesti alkuperäisen Neo Geo -version kanssa. Pallojen ammuskelu on tarkkaa, toimivaa ja viihdyttävää – kaksinpelinä tietenkin koukuttavaa, haastavaa ja riemukkaan kilpailullista.

Onhan *Puzzlessa* yksinpelikin, mutta se on ollut aina sivuseikka: hätävara, jota pelataan, kun kaveria ei ole paikalla. Ehkä se on huijaustakin – eihän tätä saa treenata kaverilta salaa!

Lataa peli: csdb.dk/release/?id=213965



COMMODOROLYMPICS

Decathlon on lähemmäs 40 vuoden jälkeenkin yksi kovimmista retrourheilupeleistä. Sen yksinkertainen rämpytystoteutus on niin toimivaa, että peliin pystyy koukuttamaan aivan kenet tahansa.

Artline Designs -tiimin tuore versiointi vanhasta klassikosta on yhtaikaa sekä veikeä että vähän turha. Se on peliksi kasvanut koodausharjoitus, joka lainaa rajusti **David Cranen** olympiaklassikosta. Se on *Decathlon*, pienillä fiilistelyeroilla: ihmispeluri kisaa C64:n kasaripeleistä ripattujen hahmojen kanssa ja menoa höystetään retrojipoilla. Huvittaahan se, kun maaliin saapuva urheilija käristetään laserilla *Impossible Mission* -tyyliin.

Rämpytys on viihdyttävää, mutta pelin kamppaa yksi puuttuva asia: olympialaiset. Lajeja pelataan vain yksi kerrallaan, monen pelaajan olympiadia ei voi järjestää, ellei tuloksia excelöidä pelin ulkopuolella. Lisäksi urheilulajeista on pudotettu veke korkeus- ja seiväshyppy, mutta kyllähän kahdeksalla pärjää, kun mukana on 1500 metrin ranteentappaja.

Lataa peli: csdb.dk/release/?id=213963

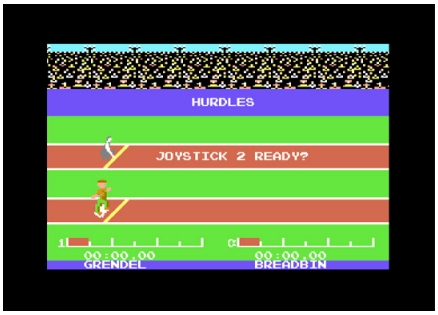
SCRAMBLE INFINITY

Vuonna 1981 julkaistu *Scramble* on yksi videopelien esi-isistä. Se ei ole yhtä tunnettu kuin moni muu aikalaisensa, mutta sen vaikutus räiskintäpeligenren syntyyn ja kehitykseen on kiistämätön: vasemmalta oikealle horisontaaliskrollaavat shootterit ovat hälle kaiken velkaa, aina *Gradiuksesta R-Typeen* ja *Parodiusiin.*

Minulle *Scramble* ja sen kaltaiset varhaispelit ovat nuoruuttani. Jotain tällaista hinkkasin omalla kuusneloselleni huoneessani. Siksi paluu sen maailmaan uuden *Infinityn* ohjaimissa on tunteikas kokemus. Tämä on niin nähty – mutta tuttuus on turvaa ja laatua. *Scramblen* yksinkertaisessa etenemisessä, räiskeessä, luolastoissa ja alati tyhjenevässä polttoainetankissa on jotain primitiivisen vetoavaa.

Ei vähiten siksi, että *Infinity* on hienosti rakennettu uusi ja kaunis tulkinta vanhasta klassikosta. Siinä missä kasari-*Scramblet* ovat... no, aikansa tuotteita, *Infinity* on nykyaikaisen värikäs ja teknisesti toimiva, vaikka pelimekaniikka onkin vanhaa koulua. Se on myös hieno tapa juhlia unohduksiin vaipuneen klassikon 40-vuotista uraa.

Lataa peli: csdb.dk/release/?id=212252



Kehittäjä: Acied/JHB (Taito)
Julkaisija: - (Taito
Vuosi: 2022 (1994)
Alusta: C64 (arcade, Neo Geo CD, lukemattomia muita)

Kehittäjä:
Mutetus / Artline Designs
Vuosi: 2022
Alusta: C64

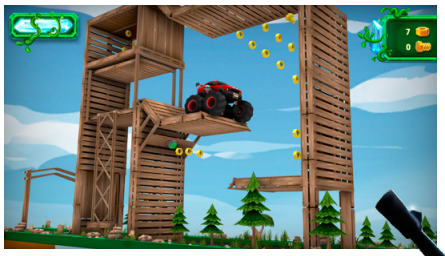
Kehittäjä:
5 O'Clock Software, Booze Design
Vuosi: 2021
Alusta: C64

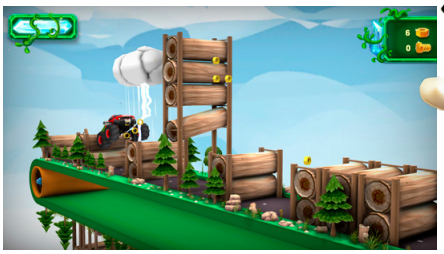
ROLLING ADVENTURE

Yksinkertainen, mutta silti kinkkinen takaviistosta kuvattu autoseikkailu kuulosti kivan retrolta uutuuspeliltä ja sitähän se tietyllä tapaa onkin. Idea on klassinen: pöristellään autolla haastavaa temppurataa pitkin kohti maalia. Sitähän *Trialsitkin* ovat.

Mitä tahansa ideaa voi silti aina väritää. *Rolling Adventuressa* auto putputtaa pysähtymättömällä liukuhihnalla, jolle peli luo maisemia proseduraalisesti. Jos pysähdyt miettimään liian pitkäksi aikaa, putoat hihnalta. Jos ajat liian lujaa, putoat hihnalta.

Pikku kiihdytyksiin ja komeisiin loikkiin pystyvällä autolla kikkailu on kivaa. Peli nousee sisällöllisesti monta askelta mobiilimaailmalla suosittujen yhden täppäyksen simplestonien yläpuolelle, yhdistellen mukavasti niin helppoja kuin tarkkoja ajoituksia ja loikkia vaativia haasteita. Pelattavuus ja vaikeustaso sopivat oivasti retron harrastajille, jotka saavat *Rolling Adventuresta* kivoja säväreitä muinaisista, sivusta kuvatuista ajopeleistä.





Kehittäjä ja julkaisija
Jumit
Vuosi: 2021
Alusta: PC

SAM & MAX: BEYOND TIME AND SPACE

LucasArts teki vuonna 1993 legendaarisen seikkailupelin mainiosta *Sam & Max* -sarjakuvasta – ja Telltale Games loi oman tulkintansa parikymmentä vuotta myöhemmin. Lucasin luomus on unohdettu pelihistoriaan, mutta Telltalen 14 vuoden takainen seikkailupelisarja on saanut erinomaista retrorakkautta. Nyt sen toinen tuotantokausi on remasteroitu nykyalustoille kaikkien viiden jaksonsa kera.

Sam & Maxin remasteroidut seikkailut ovat täynnä letkeää sanahelinää ja vimmaisen omituisia ideoita. Remasteroinneissa pelejä on tuunattu rakkaudella niin sisällön, ulkoasun kuin kohtausten ohjauksen suhteen. Myös hahmomalleja ja ohjausta on työstetty huolella uusiksi sarjakuvan luoneen **Steve Purcellin** ohjauksessa.

Seikkailuissa on myös säädettävissä oleva vihjejärjestelmä, jonka ansiosta vinkeitä puzzleja ja hervottomia tarinoita voi takoa haluamallaan vaikeustasolla, eikä matka jää keltään kesken. Enkä voi olla rakastamatta tapaa, jolla yksittäiset seikkailut nivotaan yhteen saumattomaksi kokonaisuudeksi.

Kehittäjä ja julkaisija
Skunkape Games (Telltale)
Vuosi: 2021 (2007-2008)
Alusta: PC, X1, Switch
(PC, X360, Wii, Mac, PS3, iOS)

CIRCUIT SUPERSTARS

Circuit Superstars puraisee modernilla tavalla esimerkiksi *Slicks'n'Sliden, GeneRallyn* ja *Skidmarksin* pelaajiin tarjoten lystiä isometrisestä kuvakulmasta kuvattua monipuolista ja myös monen pelaajan ajettavaa. Hupia piisaa niin yksittäisistä pikkukisoista kuin kunnon uratilallisesta ajelusta 13 eri autoluokassa, rallikaaroista ja formuloista rekkoihin.

Pelissä onnistuttu panostamaan ydinasioihin: se ei ole liian koreileva tai ulkokultainen. Visuaalit miellyttävät silmää ja radat on rakennettu selkeiksi. Kaltaiselleni täysamatöörille ne ovat väliin liiankin kapeita, mutta harjoittelulla auto lopulta pysyy suurin piirtein radalla. Ajotuntuma on oivallinen: eri autotyypit käyttäytyvät eri tavoin, mutta ovat selkeästi hallittavissa, kun niiden ominaisuudet vain koeajaa. Kontrollitkaan eivät ole liian herkät tai jäykät, vaan justiinsa kunnolliset.

Tekoäly on tosin kehnoimmillaankin todella nopea ja aggressiivinen, joten kisojen voittamiseen tarvitaan lähes täydellisiä kierroksia. Etenkin kaverien kanssa samalla/jaetulla ruudulla ajettuna kelpo iloittelua.

Kehittäjä:
Original Fire Games
Julkaisija: Square Enix
Vuosi: 2022
Alusta: PC, PS4, X1, XBX|S

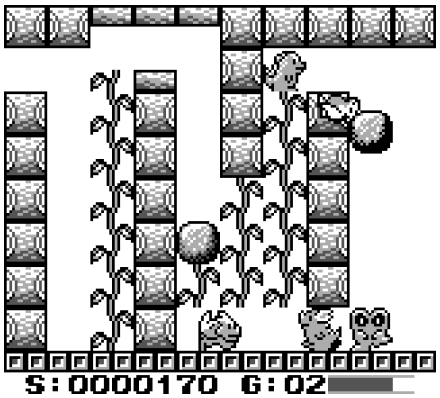
BLADES OF STEEL

Konamin kolikkopelistä koteihin kääntynyt *Blades of Steel* on melko legendaarinen kiekkoilupeli ja varsin hauska pelattava jopa tällaiselle, jota ei itse jääkendo varsinaisesti kiinnosta. Kunnon spritelätkä kuitenkin kiinnostaa aina. *Blades of Steel* nähtiin kotona ensin NES-Sillä, mutta päätyi sitten useammalle muullekin alustalle, myös taskukokoon sovitettuna Game Boy -versiona.

NES-versioon verrattuna Game Boyn terästerät on hyvin uskollinen – voisi jopa sanoa, että värit pois lukien melkeinpä yksi yhteen kopio isommasta veljestä. Toiminta on samalla tapaa hektistä eikä säännöistä liiemmin piitata tässäkään. Turpaan vetäminenkin on samalla tapaa mukana ja ihan yhtä hauskaa. Ja mikä tärkeintä, maaleja pystyy yhä tekemmään omalta kenttäpuoliskolta asti niin kuin kuuluukin!

Game Boy -peliksi *Blades of Steel* näyttää melko hyvältä, eikä värejä oikeastaan edes jää kaipaamaan. Pientä sprite-välkettä on paikoin havaittavissa, mutta ei häiritsevyyteen asti. Pelissä on myös pieniä välipätkiä esimerkiksi maalien jälkeen, joita en muista NES-versiossa olleen. Kaikkiaan peli yllättää olemalla hyvä versiointi *Blades of Steelistä*, ja jos piti NES-versiosta niin pitää luultavasti myös tästä.

Kehittäjä: Ultra Software
Julkaisija: Konami
Vuosi: 1991
Alusta: Game Boy



GODZILLA

Jos olet pilvenpiirtäjän korkuinen jättikokoinen hirviö, mikä on parasta elämässä? Tietysti työnnellä palikoita ja ratkoa puzzleja. Game Boylle julkaistu *Godzilla* onkin peli, joka ikään kuin vetää maton jalkojen alta eikä tarjoa sellaista kokemusta, mitä pelaaja ehkä mielessään odotti. Game Boy -versio perustuu osittain muunnellusti MSX-versioon.

Tavoitteena on söpöllä pikku chibi-Godzillalla työnnellä lohkareita seiniä vasten ja hakata niitä nyrkillä tohjoksi. Kun lohkareet on tuhottu, ilmestyy kentälle eri suuntaan sojottavia nuolia, joilla voi valita mihin kenttään menee seuraavaksi. Päätavoite on pelastaa labyrinttiin vangittu oma poikanen Minilla.

Kentissä on tietysti Godzillan pahamaineisia ja lähes yhtä legendaarisia kaijuja, mutta kieltämättä vaatii hieman mielikuvitusta, jos ruudulta haluaa tunnistaa Ghidorahin ja muut. Musiikki on todella pirtsakka tipitii-tasoista pimputusta, joka kyllä syövyttää reiän päähän alta aikayksikön.

Peliä voisi ehkä vähän termiä venyttäen sanoa toiminta-puzzleksi, mutta ei siinä oikein ole suuremmin toimintaa ja puzzlettelukin on vähän siinä ja siinä. Ei peli täysin onneton ole, mutta ei todellakaan sitä mitä nimen ja kansikuvan perusteella voisi odottaa.

Kehittäjä: Compile
Julkaisija: Toho
Vuosi: 1990
Alusta: Game Boy

JUSTICE LEAGUE: TASK FORCE

Kamala Darkseid on taas hyökännyt rakkaalle planeetallemme, mutta onneksi DC:n *Justice Leaguen* sankarit ovat täällä. Tämän one-on-one-tappelupelin on kehittänyt Sunsoft, jonka apuna tässä SNES-versiossa on ollut mukana myös Blizzard. Koska monet muut Blizzardin SNES-tuotokset ovat olleet mieleeni, oli tämä yksi syistä miksi pelistä aikoinaan kiinnostuin.

Teoriassa asetelmat ovat siis kunnossa, mutta peli jättää monelta osin toivottavaa. Graafisesti peli vaikuttaa kuvien perusteella asialliselta, mutta pelatessa huomaa, että animaatiot ovat hyvin vaisusti toteutettuja. Kontrollit ovat myös tarkat erikoisliikkeiden suhteen, eikä niissä onnistuminen ole niin helppoa kuin haluaisi, vaikka liikkeet sinällään toteutetaankin tyypillisillä puolikaarilla yms.

Hävettää myöntää, mutta perusasetuksilla tietokonevastustaja lakaisi minulla lattiaa niin, että hyvä jos pari iskua sain itse sisään. Helpoimmalla vaikeustasolla testattuna sitten peli taas muuttui todella pitkästyttäväksi. Ehkä joku lajityyppiin perehtynyt saisi tästä vähän enemmän irti, mutta vaikea tätä on suositella kun SNES:lle löytyy kuitenkin *Street Fighter 2*:t, *Mortal Kombatit* ja niin edelleen.

Kehittäjä: Blizzard
Julkaisija: Acclaim
Vuosi: 1995
Alusta: SNES

PHALANX

Mainitaan alkuun, että Kemcon *Phalanxin* USA-version kansikuva on paras ja hienoin maailmassa koskaan ja ikinä. "Ei voi muuta sanoa kun hattua nostaa" heille, jotka sen ovat toteuttaneet. Ja kun tärkein on saatu pois alta, kerrottakoon, että peli itsessään on melko keskinkertainen sivulle vierivä shmuppi. Se ei ole mikään huippu, mutta ei toisaalta huonokaan.

Genren kliseistä poiketen *Phalanxissa* pelaajan alus kestää kolme osumaa ennen tuhoutumista, vaihtaa lennossa kolmen eri nopeuskonfiguraation välillä ja voi kantaa mukanaan kolmea eri asetta. Ruudun tyhjentävä pommi uhraa käytössä olevan aseen.

Kontrollit ovat melko hyvät. Hidastuksia ei esiinny niin paljon kuin SNES-räiskinnöissä keskimäärin, mutta toki välillä kuitenkin. Ensimmäisessä kentässä on melko hienoa syvyyttä tuova parallaksivieritys. Eikä peli muutenkaan ole varsinaisesti ruma, vaikka ehkä graafisesti vähän persoonaton onkin. Musiikki on turhan energistä ja iloista. Ensimmäinen kenttä menee kuolemitta, mutta sen jälkeen vaikeustaso kohoaa. Kokonaisuutena tuntemukset jäävät kuitenkin enemmän plussan puolelle.

Kehittäjä: Zoom
Julkaisija: Kemco
Vuosi: 1992
Alusta: SNES

PRINCE OF PERSIA

Jordan Mechnerin legendaarinen peli on luonnollisesti julkaistu myös Nintendon klassikoille. Alkuun olin skeptinen, mutta yllätyksekseni PC-versiota rinnalla testaillessa ja muistia virkistäessäni huomasin NES-version vastaavan melko hyvin esikuvaansa. Luonnollisesti grafiikan yksityiskohdissa ja tarkkuudessa vähän menetetään toki. Kaikki on kuitenkin tunnistettavaa. Äänipuoli ei myöskään paremmille laitteille vedä vertoja.

Kontrollit pelissä ovat hyvin tönköt, mutta niinhän ne ovat alkuperäissessäkin versiossa. NES-versiossa on kuitenkin ehkä vielä hieman enemmän viivettä, joten vaatii jonkin verran totuttelua ennen kuin ne iskostuvat tarpeeksi hyvin selkärankaan. B:stä Prinssi hipsuttaa ja A:sta tekee loikan. Taistelu hoidetaan nappeja hakkaamalla.

En ole mikään mestari sen enempää alkuperäisessä pelissä kuin tässäkään, joten vaikeustason ja tiukan aikarajan takia eteneminen tyssäsi melko pian. Ilmeisesti osa alkuperäisistä kentistä on jaettu NESillä useampaan osaan ja ruuduissa on hitusen enemmän skrollaamista, mutta muuten käännös vaikutti melko onnistuneelta ottaen alusta huomioon. Jos kunnolla pelata pitäisi, valitsisin kuitenkin ehkä jonkun toisen version, mutta ei tämäkään surkealta vaikuttanut.

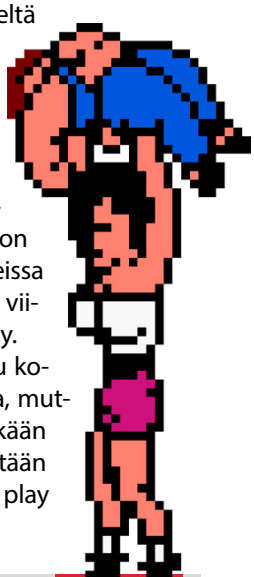
Kehittäjä ja julkaisija: Virgin Games
Vuosi: 1992
Alusta: NES

TARGET: RENEGADE

Target: Renegade on vain kotikoneille julkaistu jatko-osa Taiton beat 'em up -arcade-pelille *Renegade*. Kotitietokoneiden versiot julkaisi Ocean oman Imagine-merkkinsä turvin, mutta NES-version on julkaissut jopa ihan Taito itse. Olisi voinut kyllä jättää julkaisematkin.

Genren perinteiden mukaisesti lähdetään taas pelastamaan kaapattua henkilöä, joka ei tällä kertaa tosin ole tyttöystävä vaan veli. Totutusti pelaaja astelee kaupungilla potku- ja lyöntinapit valmiina. Hyppääminen tapahtuu painamalla ylöspäin ja potkua yhtä aikaa, ja hyppypotku onkin melko tärkeä taito taistelun tuoksinassa. Kontrollit ovat mukiinmenevät, mutta parempiakin on NESillä tullut kyllä eteen.

Ensimmäisessä kentässä vastaan tulee vain yhden ja saman ukon kopioita. Variointia on se, että välillä tuo ukko tulee moottoripyörällä päin, jolloin se pitää ensin potkia sieltä alas. Ei juuri piristä. Ekan kentän pomovastus tirvaisee kerran turpaan ja siinä menee jotakuinkin puolet täydestä energiapalkista. Elinaika on laskettavissa sekunneissa ja tässä vaiheessa viimeistään hymy hyytyy.



Kaksinpeli puuttuu kokonaan NES-versiosta, mutta en usko, että sekään olisi pelistä tehnyt yhtään parempaa. Do not play this game, dude!

Kehittäjä: Ocean
Julkaisija: Taito
Vuosi: 1990
Alusta: NES

RETROTINK 5X-PRO

PARAS SKAALAIN PARANEE YHÄ

Teksti: Mikko Heinonen
Kuva: Nanne Kukkura

Retrokoneiden sovittaminen nykyaikaisiin näyttölaitteisiin on hankalaa puuhaa. Yhä useammassa televisiossa ei ole enää ainakaan RGB-SCART-liitäntää, ja jos komposiittivideopistoke vielä löytyykin, sen kuva näyttää kohtalaisen karmealta nykyisissä jättitöllöissä. On siis syntynyt jonkinlainen markkina ulkoisille HDMI-sovittimille.

Videon käsittelyyn vaaditaan kohtuullisen paljon laskentatehoa, ja toimenpide pitäisi myös tehdä niin reaaliajassa kuin mahdollista, jotta se ei tuota kuvaan häiritsevää viivettä. Useiden kymmenien millisekuntien viive ei tule kysymykseen esimerkiksi nopeatempoista peliä pelattaessa. Tämä vaatimus sekä muutenkin monimutkainen rakenne nostavat ulkoisen skaalainlaitteen hintaa ja tekevät investointien takaisinmaksuajoista pitkiä, sillä potentiaalinen ostajakunta ei ole valtavan suuri. Monelle satunnaiselle pelailijalle kun riittää marketin hyllyltä tai Kiinan krääsäkaupasta mukaan poimittu edullinen sovitin, jolla saa kuvan jotenkin näkyviin.

Laatutietoisen kuluttajan valinta olikin pitkään helppo, sillä saatavana oli käytännössä vain yksi kunnollinen laite: Micomsoft XRGB mini Framemeister. Useamman sadan euron hintaisen, Japanista kotoisin olevan Framemeisterin laiteohjelmisto on kypsä ja se muuntaa signaalin RGB:stä, komposiitista, S-Videosta ja komponentista 1080p-tarkkuuden HDMI:ksi siedettävän nopeasti, mahdollistaen myös sen monipuolisen säätämisen. Huonoa siinä on käytännössä vain näyttötilan vaihdon hitaus: siirtyminen vaikkapa lomitetun ja lomittamattoman kuvan välillä saattaa kestää useita sekunteja, eivätkä esimerkiksi demot näy kokonaan. Kaltaiselleni pelivideoita tallentavalle tämä aiheuttaa aivan erityistä kärsimystä, sillä tallennus katkeaa aina tahdistussignaalin hävitessä.

Mikään ei kuitenkaan kestä ikuisesti, ja Framemeisterin valmistus on loppunut jo vuosia sitten, kun siinä käytetyt piirit poistuivat tuotannosta. Loppuaikoina sen hinta ehti kivuta huomattavan korkeaksi, sillä suoraa korviketta ei ollut heti tarjolla. Suomalaisen **Markus Hiienkarin** suunnittelema OSSC eli Open Source Scan Converter on edullisempi ja etenkin nopeutensa puolesta Framemeisteria parempi, mutta käytettävyydeltään kömpelömpi eikä yhtä yhteensopiva erilaisten käyttötarkoitusten kanssa. OSSC ei ole varsinainen skaalain vaan niin sanottu linjatuplain, joka käsittelee signaalin juova kerrallaan ja "kertoo" sen sopimaan television toistoalueelle. Se ei sisällä puskurimuistia, jonka avulla kokonaisten framejen käsittely olisi mahdollista, eikä myöskään osaa esimerkiksi pakottaa signaalia tiettyyn kuvasuhteeseen.

Markus on kehittämässä OSSC:stä uutta versiota, joka todennäköisesti ylittää Framemeisterin joka mittapuulla, mutta valitettavasti OSSC Pro jäi koronaviruksen aiheuttaman komponenttipulan jalkoihin ja viivästyy alkuperäisestä aikataulustaan. Sitä odotellessa markkinoille ennätti **Mike Chin** suunnittelema RetroTINK 5X-Pro. Siinä missä aiemmat RetroTINKit olivat lähinnä yksinkertaisia HDMI-sovittimia ilman erityistä kuvan jälkikäsittelyä, 5X-Pro pyrkii olemaan täysimittainen skaalain.

Tein itse ja säästin

Kuten todettua, pelikäyttöön tarkoitettujen skaalainten markkinat eivät ole valtavan suuret ja niiden kehittäjiltä vaaditaan riskinottoa ja monipuolista osaamista. Jokin asia jää tällöin väistämättä lapsipuolen asemaan, ja RetroTINKin tapauksessa se oli käyttöliittymä. Kun sain laitteen käsiini viime syksynä, sen käyttö ja asetusten tekeminen oli sanalla sanoen hankalaa. Valikoissa liikkuminen oli kankeaa ja koska muutokset otettiin heti käyttöön, oli hyvin helppoa hävittää kuva näytöltä. Mukana toimitettu kaukosäädin on alkujaan jostain muusta laitteesta ja toimi alkuun niin hämärästi, että siitä piti liittää laitteen mukaan erillinen ohjelappunen. Tämä ei sinällään ollut mitään uutta, sillä OSSC:n mukana toimitettu yleismallin kaukosäädin tai Framemeisterin kokonaan japaninkielinen kapula synnyttivät nekin oman markkinansa jälkikäteen asennettaville ohjeteipeille.

Tilanne kuitenkin muuttui ratkaisevasti, kun juuri ennen joulua 2021 julkaistiin laiteohjelmiston versio 2.39. Se uudisti koko käyttöliittymän ja teki siitä huomattavasti järkevämmän käyttää kaukosäätimen avulla: valikko-, nuoli- ja OK-painikkeet alkoivatkin yllättäen tarkoittaa jotain. Asetukset on nyt mahdollista valita ensin käyttöön ja vahvistaa sitten OK:lla, mikä vähentää virheen mahdollisuutta, ja esimerkiksi väärän näyttötilan valitessaan ei tarvitse enää käydä läpi kaikkia mahdollisia vaihtoehtoja sokkona. Lyhyesti voi todeta, että 2.39 teki laitteesta kypsän myytäväksi muillekin kuin erittäin teknisesti suuntautuneille.

On silti helppo nähdä, että RetroTINK 5X-Pro on piensarjana valmistettava tuote. Varsinaisen pakkauksen sijaan se toimitetaan sympaattisessa pussukassa, eikä mukana seuraa esimerkiksi muun-

tajaa vaan pelkkä USB-johto. Laiteohjelmiston päivitysohjeessa lukee erikseen, ettei kannata huolestua, jos päivitysohjelma lakkaa vastaamasta parin minuutin ajaksi. Useamman sadan euron hintaisen laitteen kanssa tämä ominaisuus ei aina ole omiaan laskemaan pulssia, vaikka pieleen mennyt ohjelmointi pitäisikin olla mahdollista palauttaa melko helposti.

Kattavat ominaisuudet

Kun unohdetaan jokseenkin ymmärrettävä kotikutoisuus ja keskitytään puhtaasti tekniikkaan, RetroTINK 5X-Pro on hyvin vakuuttava tuote. Se pystyy ottamaan vastaan hyvin monenlaisia signaaleita ja muuntamaan ne HDMI:ksi jopa 1440p-tarkkuudella. Liittimiä löytyy RGB:lle, komposiitille, S-Videolle ja komponentille, ja laite ottaa myös äänet mukaan HDMI-signaaliin. Tähän liittyy tosin pieni kauneusvirhe, sillä toistaiseksi äänen tulotasoa ei voi säätää kuten Framemeisterissa ja OSSC:ssä. Vaikka linjatasoisen audiosignaalin pitäisi periaatteessa olla aina samanlaista, käytännössä näin ei ole, ja tämä aiheuttaa ainakin itselleni jatkuvia haasteita videoiden tallennuksessa.

Vastapainona sitten RetroTINK 5X-Pro on ensimmäinen skaalain, joka selviää esimerkiksi tietyistä Amigan skenedemoista tai monista Sega Saturnin ja Sony PlayStationin peleistä ilman signaalin jatkuvaa pätkimistä – ne kun tykkäävät räpsytellä näyttötilaa lomitettuun ja takaisin tuon toista. Joitain satunnaisia yhteensopivuusongelmia on toki tullut vastaan, mutta pääosin RT5X on korvannut omassa käytössäni Framemeisterin 90-prosenttisesti ja tietyille laitteille kokonaan, mikä ei ole lainkaan vähäinen saavutus. Hinta on korkeahko, mutta laadukasta, viiveetöntä ja monipuolista skaalainta etsivälle muita vaihtoehtoja ei käytännössä ole – ainakaan ennen OSSC Pron ilmestymistä.

RetroTINK 5X-Pro
Kehittäjä: Mike Chi
Hinta: 300 $ + rahdit + verot
Verkkokauppa: www.retrotink.com

MARSEILLE MCLASSIC

ULKOINEN NÄYTTÖKORTTI RETROKONEESEEN?

Toimitukseen tarjottiin testiin kiinnostavan kuuloista laitetta, jonka luvattiin päivittävän vanhempien pelikonsoleiden grafiikan nykyaikaan.

Teksti: Mikko Heinonen

Mainospuheen mukaan HDMI-liitäntään kytkettävä, noin 130 euron hintainen Marseille mClassic jälkikäsittelee ja skaalaa kuvasignaalia yksi frame kerrallaan, silittäen etenkin vanhojen 3D-pelien rosoisuutta. Takakansi luettelee pitkän rimpsun konsoleita, joiden kuva paranee heittämällä, ja mainostaa laitetta kuin näytönohjainpäivityksenä konevanhukseen. Facebook-mainosten mukaan laite on saanut käyttäjiltään erinomaiset arviot.

Puhutaan nyt ensin siitä, mikä on mahdollista. HDMI-kuvaa voi tosiaan jälkikäsitellä monin tavoin, mutta kun kyseessä on pelikuva, ei käsittelyyn saisi kulua kovinkaan kauaa – mieluiten ei lainkaan ylimääräistä aikaa. Ja kun prosessointia tekee yksittäinen ASIC-piiri, joka ilmeisesti kuluttaa vain muutaman watin sähköä, ei operaatio voi olla kovinkaan monimutkainen.

Ongelmaksi muodostuu myös se, että signaalin pitäisi olla jo valmiiksi HDMI:tä, mikä rajaa pois vanhimman pään konsolit. Testasinkin mClassicia ensin RetroTINKin kanssa skaalaamalla PlayStationin RGB:n 1080p-tarkkuuteen. Erot kuvanlaadussa olivat marginaalisen pieniä, jotain pientä siistiytymistä olin näkevinäni väripinnoissa ja ääriviivoissa, mutta ei millään 130 euron arvosta. Sitten kokeilin Dreamcastin VGA-signaalia (640x480) suoraan HDMI:ksi muunnettuna, enkä saanut laitetta tekemään sille oikein mitään. Ainoat säädöt kun ovat pois päältä, käsittelytila päällä ja erityinen retrokäsittelytila.

Otin asian puheeksi Facebookissa ja sain vihjeen, että mClassic toimii parhaiten, kun se saa itse käsitellä 720p-signaalin 1080p-muotoon. Ilmeinen käyttökohde olisi siis PlayStation 3, jonka peleistä suuri osa pyörii 720p-tarkkuudella. Vaan tässäpä tulikin sitten mutka matkaan vertailun suhteen: PS3:ssa on rautatasolla HDCP-kopiosuojaus, joten siitä ei tallenneta yhtään mitään, ellei HDCP:tä kierrä käyttämällä sopivaa HDMI-jakajaa, joka taas oli hukassa. Yksi PS3:n omistaja kuitenkin kertoi huomanneensa eron ennen ja jälkeen mClassicin, joten uskotaan. Hän oli ostanut laitteen n. 80 eurolla Kickstarter-kampanjasta, eikä olisi itsekään ollut valmis sijoittamaan siihen 130 euroa.

Ei mClassic siis mikään huijaus ole, mutta se vaatii toimiakseen täsmälleen oikeat olosuhteet sekä pelin, joka käyttää 3D-grafiikkaa juuri sopivan rosoisesti. Bittikarttagrafiikalle parantelusta ei ole mitään hyötyä, lähinnä haittaa. Myyntipuheisiin ulkoisesta näyttökortista kannattaa siis suhtautua suurin varauksin.

ATARI 2600 TALVISODASSA

MUUTAMAN KILOTAVUN TARKKA-AMPUJA ITÄRAJALLA

Teksti: Manu Pärssinen
Kuvat: Wolfgang Kuhmann

Viime vuoden lopulla Facebookin AtariAge-ryhmää rullatessa esiin vyöryi kuva, joka pysäytti: siinä oli selvästi Atari 2600 -pelimoduuli ja aivan yhtä selvästi pelin etiketissä luki selvällä suomella "Talvisota". Pieni alustava googlailu paljasti, ettei peli ollut kotimaista alkuperää, joten mysteeri oli valmis – ja sehän piti selvittää!

Talvisota-pelin tekijä Wolfgang "donnerkuh" Kuhmannin hyllyt ovat täynnä miehen Atari-kokoelman helmiä.

Onnekseni maailman retropiirit ovat lopulta melko pienet ja kävi ilmi, että tätä kotikutoista *Talvisota*-peliä myy tämänkin lehden sivuilta tutun **Michael Thomassonin** Good Deal Games -verkkokauppa. Hänen kauttaan onnistuin hankkimaan kopion pelistä itselleni sekä ottamaan yhteyden pelin tekijään, joka keskusteluryhmissä esiintyy nimellä "**donnerkuh**".

Atari 2600:n *Talvisodassa* pelaaja on tarkka-ampuja (ehkä jopa itse **Simo Häyhä**), jonka tehtävänä on pidätellä Neuvostoliiton armeijaa mahdollisimman pitkään, jotta siviilit pääsevät pakenemaan. Pelin pisteytys ei tulekaan ammutuista vihulaisista vaan siitä, montako ihmistä on päässyt etenevän armeijan tieltä turvaan.

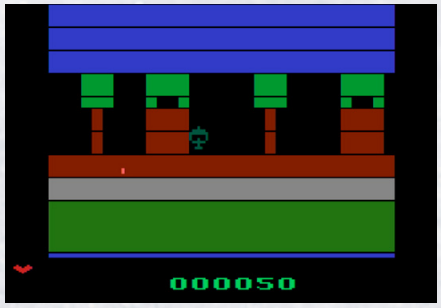
Joystickilla ohjataan käsissä tärisevää ampujan tähtäintä. Mikäli pyssyttelee metsään ja vihollinen jää henkiin, ampujasankarimme hermostuu, pulssi nousee ja tähtääminen muuttuu entistä hankalammaksi. Pelimekaanisena ominaisuutena panssarivaunujen ampumisella voi rauhoittaa pulssia (ehkä tämä toimii oikeassa elämässäkin?), mutta mikäli syke nousee huippuunsa, on pelaajan pakko heiluttaa valkoista lippua ja peli päättyy.

Vihollisia ei ole vain yhtä sorttia. Bunkkeriin pyrkivien sotilaiden lisäksi ruudulla näkyy piilostaan räiskiviä konekivääreiä, varusteita vihollisille kuljettavia kuorma-autoja sekä jopa lentokoneita, jotka nekin voi taitavalla laukauksellaan tiputtaa. Graafisen ulkoasun perustaso selittyy osittain sillä, että peliä ei ole ohjelmoitu raudan kikkoja assemblerilla äärimmilleen puristaen, vaan **Fred Quimbyn** kehittämällä *batari Basic* -ohjelmointikielellä. Sen avustamana Atari 2600 -tuotosten tekeminen on kohtuullisen yksinkertaista – ainakin yksinkertaisempaa kuin 1980-luvulla.

Mikään Atari 2600:n kykyjen huippunäyte *Talvisota* ei ole, mutta kyllä siitä löytyy jonkin verran sitä yksinkertaista varhaisten retropelien koukkua, joka saa yrittämään uudelleen korkeampaa pistetulosta. Onhan pelaajalla kuitenkin Suomen kunnia puolustettavanaan!

Haastattelussa donnerkuh

Nimimerkin donnerkuh takana on Saksan Klievessä asustava **Wolfgang Kuhmann**. Kuhmann on pitkän linjan Atari-harrastaja, vaikka ei koskaan lapsena saanutkaan Atari 2600:aa, vaan sitä piti mennä kavereiden luokse pelaamaan. Vuonna 1985 hän sai kuitenkin Atari 800 XL:n, jonka kanssa syntyi monia rakkaita muistoja. Vaikka PC-maailma kutsui miehen mukaansa, on hänelle



Yksinäinen pikseli, vastassa koko Neuvosto-armeija.

Donnerkuh teetti pelistä myös hyvin rajoitetun erän laatikoineen päivineen.

kertynyt vuosien varrelta kokoelma retrolaitteistoa kirpputoreja kierrellen.

Wolfgang kertoo löytäneensä AtariAge-sivustolta *batari Basicin*, joka herätti vanhat *Atari Basic* -muistot. Mies loikkasi heti Atari 2600 -pelikoodauksen maailmaan ja ensimmäisenä syntyi peli nimeltä ISS.

- Sen pelin tarkoituksena oli enemmänkin esitellä *batari Basicia* isommalle joukolle. Youtubesta löytyy saksankielinen video, jolla koodaan kyseistä peliä. Muille laitteille en ole pelejä tehnyt, mutta Atari XL -ohjelmointi kiinnostaa yhä. Työn alla on myös kaksi uutta 2600-peliä: *Swordman* ja *Trenches of War*. Niiden keskeneräiset versiot löytyvät AtariAgen keskustelualueelta.

Miten pelin aiheeksi valikoitui suomalaisille hyvin tuttu talvisota?

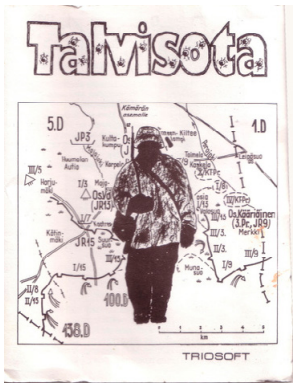
- Aluksi ajatuksenani oli vain tehdä tarkka-ampujapeli. Kun peli oli valmis, aloin etsiä sopivaa kuvaa sen etikettiin ja laatikkoon. Löysin taiteilija **Oprisnyashkan** tekemän *"Talvisota 75"*-nimisen kuvan, joka oli täydellinen ja onnekseni sain luvan sen käyttöön. Siksi peli sijoittuu nyt Suomen talvisotaan. Kuvassa oli Talvisota-teksti, joten päätin käyttää sitä pelin nimenä, enkä englannin kieltä. Myös tekemäni vihollis-spritet sopivat kätevästi siihen aikakauteen.

Wolfgang kertoo, että ohjelmointi ei ole hänelle vaikeaa – ennen kuin on aika äänitehosteiden koodaamisen. Lisäksi alusta rajoittaa ideoiden toteuttamista: Wolfgang kertoo keksivänsä peleihinsä aina lisää ja lisää ominaisuuksia, kunnes *Stella*-emulaattorin rajat tulevat vastaan ja pelin sisältöä pitää alkaa karsia. Kyseiseen ongelmaan kaatui mm. miehen hedelmäpeliprojekti *Top Slot Machine*.

- *Talvisodassa* olen ylpein tärisevästä tähtäimestä, joka tärisee sitä enemmän, mitä stressaantuneempi pelihahmo on. Se on pelin keskeinen haaste.

Tämäkään kotikutoinen peli ei syntynyt täysin yhden koodarin tekeleenä, vaan donnerkuh haluaa kiittää Thomassonia fyysisten pelimoduulien tuottamisesta, **Marc Oberhäuseria** pakkauksen kanssa avustamisesta sekä paikallista yritystä etikettien tulostamisesta.

- Seuraavaksi aion viimeistellä *Trenches of War*- ja *Swordman*-pelit. Jälkimmäinen niistä on melkein valmis - minun pitää enää ohjelmoida viimeinen vastustaja ja säätää pelin tasapainoa.



Todennäköisesti varhaisin digitaalinen talvisodinta julkaistiin MSX:lle.

TALVISOTA PELEISSÄ

Donnerkuhin Talvisota ei ole ensimmäinen kerta, kun Suomen sotahistorian tunnetuimpia tapahtumia käytetään peli-inspiraationa. Lautapeleinä aiheesta on julkaistu vieläkin enemmän, mutta digitaalisina löytyvät ainakin nämä:

TALVISOTA

MSX, 1987, Olli Kainulainen, Triosoft

Talvisotaan sijoittuva tekstigrafiikkapohjainen strategiapeli (kuvia sivun alareunassa), joka sekoittelee joukkojen liikuttelua sodan tapahtumista kertoviin uutisiin.

TALVISOTA: ICY HELL

PC, 2006, Blizfront, Nival

Kotimaista tuotantoa olevaa reaaliaikastrategiaa, joka pyrki olemaan historiallisesti mahdollisimman tarkka: se on luokiteltu myös opetuspeliksi.

ORDER OF BATTLE: WINTER WAR

PC, PS4, 2016, The Artistocrats, Slitherine

Vuoropohjaista taktista strategiaa tämäkin, mutta huomattavan laajemmassa mittakaavassa kuin MSX-versiossa.

SGS WINTER WAR

PC, Mac, 2020, The Strategy Game Studio, Avalon Digital

Strategy Game Studio -sarjaan kuuluva kampanja, johon tehtiin juuri äskettäin iso päivitys.

WINTER WAR 1939

PC, Android, 2020, Thorium Game Lab

Yhden miehen projekti, josta on myös mobiilipeliversio. Pelissä voi ohjata myös kaikkia ajoneuvoja, mm. lentokoneita.

TALVISOTA - WINTER WAR

PC, 2020, Talvisota Team, Tripwire

Täysmuutos-modaus *Rising Storm 2: Vietnam* -peliin.

TEKSTIT: JUKKA O.KAUPPINEN

ALWA'S LEGACY

Muutaman vuoden takainen *Alwa's Awakening* saa jatkoa *Alwa's Legacyssä,* joka on herkullinen metroidvania-tasoloikkaseikkailu. Ruotsalainen tiimi viimeisteli pelin rahoituksen Kickstarter-palvelun kautta ja täytyy sanoa, että rahalleen saa harvinaisen hyvää vastinetta.

Alwa's Legacy on upean laaja, monipuolinen ja mukaansatempaava tasoloikkailu, johon pitkien pelien ystävien kelpaa paneutua. Touhu on ensimetreiltään ehtaa metroidvaniaa: tuonne ei pääse, tässä on este. Joskus myöhemmin varmaan löytyy keino, jolla tästä selvitään. Varusteiden ja magian hiljalleen karttuessa tekee jo hissukseen mieli palata taaksepäin katsomaan aiemmin tavoittamattomia paikkoja, samalla kun matka etenee ongelmanratkonnan kera kauniissa ja värikkäässä fantasiamaailmassa. Ja vaikka kerronta on genrelleen tyypillisen ohutta, siitä löytää myös houkuttelevaa persoonallisuutta.

Maailma on rakennettu kauniista retropikseleistä, mutta nykypelien tekniikalla ja tyylillä. Peli näyttää historialliselta, mutta on täyttä tätä päivää kaikkine animaatioineen ja pikkudetaljeineen. Se on erinomainen esimerkki laadukkaasta uusretrosta. Teosta ei ole rajattu muinaiskoneiden rajoituksiin, vaan tekijät ovat rakentaneet valtavan maailmansa ja sen monituiset tehtävät muistia säästämättä. Laatuteos nousee ketterästi nykypäivän parhaimpien metroidvanioiden joukkoon.

Kehittäjä ja julkaisija: Elden Pixels
Vuosi: 2022
Alustat: PC, PS4, Switch

TROPHY

Uudet pelit NES-konsolille ovat aina hieno asia, mutta entäpä kun NES-peli julkaistaan natiivina Xbox One/X/S -versiona? Siinä mennään jo kohtuullisen syvälle retron merkityksen ytimeen. RR:ssä jo aikaisemmin NES-versiona arvosteltu *Trophy* saa näet nyt kyytiä myös uudemmilla alustoilla. Se tuntuu ristiriitaiselta, sillä en oikein nauttinut kasibittisestäkään versiosta. Olihan siinä reilusti *Mega Man* -fiilistelyä ja kivoja ratkaisuja, mutta se ei ollut minun pelini.

Kokonaisuus kun on murhaavan vaikea ja yksioikoinenkin, eikä sama peli uudemmalla raudalla paljon parempi ole. Onneksi isompi näyttö ja tarkemmaksi kasvatettu grafiikka sentään tekevät ranteiden viiltelystä miellyttävämpää, mutta kyllä kuvakokoa olisi saanut kasvattaa modernimmalla alustalla – neliömäinen 1:1-kuvasuhde näyttää toki historialliselta mutta kamoon, kyllä nykykonsolit jaksavat pyörittää retropeliä jopa laajakuvana. Penteleet vain ovat halunneet pitää niin tiukasti kiinni alkuperäisalustan olemuksesta ja rajoituksista, että Xbox Series X jopa nykii niissä kohdin, joissa myös NESistä loppuu teho kesken.

Äärimmäisen pitkän pinnan ja taitavan hypipompiammu-selkäytimen omistavalle *Trophystä* voi olla iloa, mutta muut eivät saa siitä kuin retropukamia.

Kehittäjä: Gradual Games
Julkaisija: 8-Bit Legit
Vuosi: 2022
Alustat: Xbox One, NES

CLOCKWORK AQUARIO

Pelialan historiasta löytyy runsaasti lähes tai täysin valmiita pelejä, joita ei koskaan julkaistu. Osa niistä saa rakkautta jälkikäteen, kuten vuonna 1992 valmistunut tasoloikka *Clockwork Aquario.* Sega hyllytti System 18 -alustalle tehdyn kolikkopelin, joka nyttemmin kaivettiin kadoksista ja julkaistiin moderneille alustoille.

Clockwork on paikoin rakennettu uudelleen, sillä osa lähdekoodeista ja materiaaleista on kadonnut vuosikymmenien mittaan. Valtaosa pelistä on kuitenkin aitoa ja alkuperäistä ysäriä, ja jälleenrakennuksessa on ollut mukana myös peliä aikoinaan tehneen Westonen kehittäjiä. Paketti onkin niin aito ja alkuperäinen kuin kertaalleen kadonnut, unohdettu ja restauroitu peli vain voi olla.

Nykysilmin nähtynä Segan päätös pelin hylkäämisestä kummastuttaa. Peli on toimiva, värikäs ja visuaalisesti veikeä toimintatasoloikka, jossa on omintakeisiakin ratkaisuja. Tahti on rauhallinen ja pelaaja voi edetä ilman paniikkikohtauksia. Kaksinpelikin löytyy. Kokonaisuus on herkullinen, joten ei voi kuin hämmästellen kunnioittaa Segan aikanaan valtavan korkeita tuotantokriteerejä – *Clockwork Aquario* kun on tämän päivän uusretron vyöryn keskellä maukas historiallinen makupala niille, jotka kaipaavat *Mario-* tai *Wonder Boy* -sarjojen tyylistä loikintaa veikeillä silmäniskuilla höystettyinä.

Kehittäjä: Strictly Limited Games (Westone)
Julkaisija: ININ Games (Sega)
Vuosi: 2021
Alustat: PS4, Switch (Arcade)

SONIC THE HEDGEHOG

YLLÄTTÄVÄ KUUSNELOSKÄÄNNE

Segan legendaarinen siili Sonic on tähdittänyt kymmeniä ja taas kymmeniä eri pelejä firman omilla ja muidenkin pelikoneilla. Aivan kaikkialle ei siilikään silti ole ehtinyt, vaikka vaihtelevan kamalia versiointeja löytyykin monelle alustalle, kuten Tiger Game.comille ja Sega Picolle. Tasavallan tietokoneella virallista siiliä ei ole kuitenkaan koskaan nähty, vaikka Zzap!-lehti kertoikin aikoinaan U.S. Goldin saaneen oikeudet kuusnepaversiointiin.

Eikä virallista käännöstä nähdä nytkään – mutta jos se juoksee kuin siili ja loikkii kuin siili, niin onko se silloin siili?

Erinomaisen taitava 6502-kooderi **Mr. Sid,** joka teki myös *Prince of Persian* C64-käännöksen, on nyt tovereineen kiillottanut kultaa ja hionut timanttia äimistyttävään retroloisteeseen. Lopputulos on peli, jota kukaan ei osannut odottaa eikä kaivata – peli, joka jysäytti kansan tuoleiltaan, kun se tulla tupsahti ladattavaksi. "Tässä olisi tällainen, olkaa hyvät."

Rehellisyyden nimissä on kuitenkin sanottava, ettei kuusnelosen *Sonic* ole se *Sonic*, jota todennäköisesti ensin ajattelette. Se ensimmäinen *Sonic* on näet olemassa kahtena eri versiona: Mega Drivellä ja Master Systemillä. Molemmat ovat aivan oma pelinsä. Tämä on käännös jälkimmäiseltä.

Seuraavaksi kannattaa muistaa, että Master System on puhtaasti pelaamiseen optimoitu pelikonsoli vuodelta 1985, jossa on muun muassa C64:ää tehokkaampi prosessori, samoin kuin kohtalaisesti rommia ja rammia pelimodulien oman ROM-muistin lisäksi. Kun huomioi, että moduuleilla saattoi olla jopa 512 kilotavua ROMia, niin kyllä siinä kuusnepan sisäinen muistikapasiteetti ja hidas levyasema jo vähän köhivät. Nepan perusspeksit ovat siis vähän vaatimattomammat, vaikka kone onkin monikäyttöisempi.

Vuosikymmenten kova neparaudan hakkaaminen tuottaa kuitenkin tuloksia. Mr. Sid on kyennyt optimoimaan Segan Sonicin kuuslankulle jotakuinkin täydellisesti. Ei minulta mitään puutu, kun tätä pelaan, ainakin jos käytössä on C64:n muistilaajennus. *Sonic* nimittäin vaatii REU-laajennusyksikön. Sellaisen voi haalia koneensa aitona Commodoren lisälaitteena (128-512 Kt) tai 1541 Ultimaten kautta (16 Mt). Myös Ultimate 64 tukee sitä. Digimmät kokeilijat taas voivat aktivoida REU-laajennuksen emulaattorissa tai The C64:ssä.

REU-muistilaajennuksesta on Sonicissa monenlaista hyötyä: se esimerkiksi mahdollistaa koko pelin lataamisen kertalaakista muistiin. REU:ssa on myös tavallista C64:ää nopeampi DMA-muistiosoitus, eli tietokone voi siirtää ja käsitellä dataa peruskonetta nopeammin. DMA-siirto ei myöskään rasita prosessoria, eli kooderille jää enemmän CPU-aikaa aikaa muun pelikoodin pyörittämiseen.

Kun tähän yhdistetään muutenkin nohevaa koodikikkailua, niin C64:n *Sonicille* ei voi kuin nostaa hattua ja laittaa portaiden viereen oikein maukkaan etana- ja raksutarjoilun. Se on niin jäätävän, odottamattoman toimiva ja nopea. Vaikka kuusnelonen onkin kelpo kone arcadepeleihin, niin *Sonicin* tekninen toteutus nostaa tekniikkarimaa ja haastaa nykypäivän pelintekijöitä oikein kunnolla yrittämään entistäkin paremmin. Näin virheettömään ja sulavaan 50 kuvaa sekunnissa -nopeuteen tasoloikinnassa ei ole moni ole kyennyt.

Luvaton, omin nokin tehty kuusnelos-*Sonic* toimii, pyörii ja kiitää. Kokemus on täysin odottamattoman täydellinen. Kasibittiversio on myös Mega Driveä rauhallisempi, setämiesmäisempi, joten siitä on helpompi nauttia kaikessa rauhassa, omaan tahtiin. Se on rehellisesti oma, toimiva pelinsä, joka tulkitsee prameampaa *Sonicia* taitavasti edellisen sukupolven raudalla.

Jos hyvin käy, niin ehkä *Sonic* osaltaan usuttaa muitakin käyttämään REU-laajennuksia peleissään. Tuskin REU-peleistä valtavirtaa tulee, mutta olisihan se valtavan hienoa, että vuonna 1986 C64/128:lle esitelty hyödyllinen lisälaite saisi viimein kunnon rakkautta ja hyötykäyttöä.

Lataa peli: csdb.dk/release/?id=212523

Kehittäjä: Mr. SID / HVSC (Ancient) Julkaisija: - (Sega) Vuosi: 2021 (1991) Alustat: C64 (Master System, Game Gear)

ERIKOISEMPIA SONIC-PELEJÄ

Teksti: Mikko Heinonen

C64:n *Sonic the Hedgehog* ei ole virallinen käännös, mutta Sega on toistaiseksi antanut sen olla jakelussa, sillä peliä ei kaupitella rahaa vastaan. Eipä noin upea versiointi toki huonoa mainosta olekaan.

Japanin pelijätti on myös itse lainannut maskottiaan useasti edistämään muiden konsoleiden menekkiä. Tulokset ovat olleet jonkin verran vaihtelevia.

Sonic Jam (1997, Tiger Game.com)

Game.com oli perin erikoinen käsikonsoli, joka yritti samalla olla PDA ja tarjosi jopa tekstipohjaisen internetyhteyden sekä ensimmäisenä pelikoneen kosketusnäytön. Yhtä hämmästyttävää oli se, miten paljon erilaisia tunnettuja nimikkeitä Tiger onnistui konsolilleen hankkimaan – oli muun muassa *Duke Nukem 3D, Mortal Kombat, Indy 500*… ja *Sonic Jam.* Saman niminen kokoelma julkaistiin samoihin aikoihin myös Sega Saturnille.

Emulaattorilla kokeiltuna Game.comin *Sonic Jam,* jota muuten pelattiin taannoin myös pikapelimaraton Games Done Quickissa, tuntuu vähän kummalliselta. Kyllä tämä ikään kuin on *Sonic*, mutta oudon kapealla näyttöalalla eikä likimainkaan niin vauhdikkaasti kuin mihin on totuttu. Molempia selittänee se, että hintalappu edellä suunnitellun Game.comin näyttö on varsin hitaasti reagoivaa vihermustamallia eikä oikeastaan sovellu kovin raivokkaaseen toimintaan.

SonicN (2003, Nokia N-Gage)

N-Gagesta on puhuttu – ja tullaan vielä puhumaan – paljon tämänkin lehden sivuilla, ja jokaisella on siitä ainakin mielipide. Nokian suurpanostus pelipuhelimeen karahti kiville monestakin syystä, mutta nimekkäistä peleistä asia ei kiinni jäänyt. Monen muun sankarin ohella myös Segan sinisiili viiletti taco-puhelimen näytöllä.

N-Gagen emulointi vaatii edelleen mielenterveydeltä aika paljon, mutta kun sen saa lopulta toimimaan, luvassa on yllätys: *SonicN* tuntuu yllättävänkin mukavalta. Itse asiassa kyseessä onkin käännös Game Boy Advancen *Sonic Advancesta* (2001), joka on varsin laadukasta siilitoimintaa. Jos unohdetaan N-Gagen pystynäytöstä johtuva hieman erikoinen kuvasuhde, pelissä ei ole mitään isompaa ongelmaa. Ja siinä se vika taisi vähän ollakin: kun sama peli oli vielä parempana GBA:lla, miksei käyttäisi sitä.

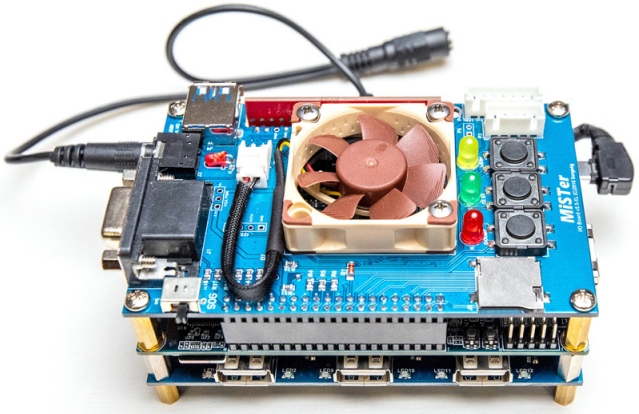
MITÄ KUULUU, MiSTer FPGA

Uudelleenohjelmoitavan FPGA-piirin avulla on mahdollista toteuttaa huomattavan tarkka kopio toisesta laitteesta. MiSTer-projekti kokoaa yhteen monien eri kehittäjien työn tuloksia.

Teksti: Mikko Heinonen

Kirjoitin Skrollin numerossa 2019.2 (lataa PDF ilmaiseksi: skrolli.fi/numerot) tuolloin melko tuoreesta MiSTer-projektista. Kyseessä on Terasic DE10-Nano -opetuspiirilevyyn ja sillä majailevaan Altera Cyclone V -FPGA-piiriin perustuva toteutus, joka pyrkii tavallaan yhdistämään monet erilaiset hankkeet vanhojen kotimikrojen, pelikonsoleiden ja peliautomaattien mallintamisesta ohjelmoitavien piirien avulla. Sen sijaan, että jokaiselle laitteelle pitäisi hankkia oma uusiopainoksensa, MiSTer on kuin kameleontti, joka mukautuu kulloiseenkin käyttötarkoitukseen. DE10-Nanon ympärille voi joko rakentaa itse tai ostaa valmiina erilaisia SDRAM-muistipiirejä sekä I/O-kortteja, joilla siihen voi liittää näppäimistön, peliohjaimia ja paljon muuta.

Kolme vuotta myöhemmin yhteisö laitteen ympärillä on vain kasvanut entisestään. Mystisen nimimerkki **Sorgeligin** takaa on paljastunut **Alexey Melnikov,** joka tekee laitteen parissa aktiivista kehitystyötä, ylläpitää projektia ja Linuxin päällä toimivaa käyttöliittymää, julkaisee uusia emulaatioytimiä eli *coreja* ja päivittää entisiä. Lisäksi mukana on suuri joukko muita aktiiveja, jotka työstävät joko yksittäisiä, monimutkaisempia laitteita tai esimerkiksi tietyn arkkitehtuurin kolikkopelejä. Useimmat heistä toimivat Patreon-palvelun kautta, jolloin työn tukijat saavat kokeilla beta-versioita ja vaikuttaa kehityksen suuntaan, mutta lopulta kaikki kuitenkin tulee myös julkiseen jakoon. Itse olen Melnikovin lisäksi seuraillut ja tukenut espanjalaista **Jose Tejadaa**, joka puuhastelee erilaisten kolikkopeliytimien, kuten Sega System 16:n ja Capcom CPS:n parissa.



Kytke ja käytä

Monesta muusta vastaavasta avoimen lähdekoodin hankkeesta MiSTerin erotti jo alun pitäen se, miten helppoa käyttöönotto oli – etenkin huomioiden sen, että laite on käytännössä vain yhteen nidottu kasa sekalaisia piirilevyjä, ellei sitten erikseen halua hankkia sille koteloa. Tämä on edelleen totta vuonna 2022, ja kaikki on vain parantunut. Kirjoittamalla netistä löytyvän levykuvan SD-kortille saa helposti käynnistettyä automaattisen asennuksen, joka noutaa halutut tiedostot suoraan netistä käyttämällä DE10-Nanon Ethernet-porttia tai USB:llä liitettyä langatonta verkkokorttia. Pienellä googlaamisella löytää vieläkin laajempia komentosarjoja, jotka paitsi päivittävät kaikki koneen emulaatioytimet, myös noutavat niille pureskeltavaksi ROM-tiedostoja mm. Archive.org-palvelusta. Käyttäjän ei tarvitse kuin kopioida skripti kortille, painaa Enteriä ja jäädä odottelemaan (hyväksi toviksi, sillä latausnopeus ei aina päätä huimaa). Update_all-skriptille voi myös kertoa, keitä kehittäjiä tukee rahallisesti, jolloin heidän beta-tuotoksensa pysyvät myös ajan tasalla yhtä vaivattomasti.

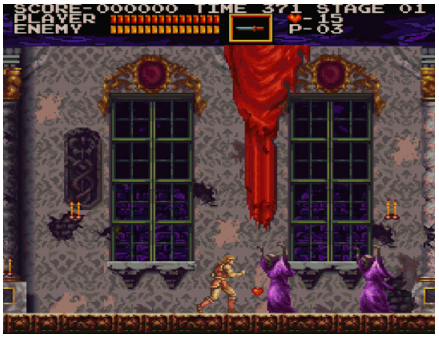
Olikin siis varsin helppoa palata MiSTerin maailmaan, vaikka kone oli ollut jonkin aikaa käyttämättä ja olin vahingossa tullut alustaneeksi siinä käyttämäni 128 gigatavun microSD-kortin toista tarkoitusta varten. Siirryinkin saman tien tuplakokoiseen, sillä olin kuullut, että käyttökelpoisia järjestelmiä on tullut paljon lisää. Valikoimaa tosin rajoittaa hieman se, että omassa MiSTerissäni on vain 64 megatavun SDRAM-muisti, kun osa ytimistä käyttää jo 128 megaa CD-levykuvien lataamiseen. Muistinlaajennuksen voisi joko tehdä kohtuullisen helposti itse tai ostaa valmiina noin 50 euron hintaan, mutta vuoden 2022 logistiikalla kumpikaan ei suju kovin nopeasti, joten on tyytyminen nykytilaan. Silläkin uutta tutkittavaa riittää vaikka miten.

Terveisiä kotimaastani Japanista

Olen myös Retro Rewindin sivuilla hehkuttanut usein japanilaisia kotimikroja sekä niiden upeita pelejä. Etenkin Sharp X68000:n hienoihin kolikkopelikäännöksiin tutustuminen on tähän mennessä vaatinut kuitenkin joko merkittävää investointia alkuperäiseen rautaan tai sitten vanhan ja käyttöliittymältään kohtalaisen hämärän emulaattorin säätämistä toimintaan.

MiSTer laskee kynnystä huomattavasti, sillä edellisen tutustumiskerran jälkeen X68000-ydin on ottanut isoja harppauksia eteenpäin. Archive.org-palvelusta löytyy valmis paketti HDF-tiedostoja, joihin ystävällinen harrastaja on pakannut koneen parhaita pelejä. Toistaiseksi ydin tukee ainoastaan näitä pieniä SASI-levykuvia, joiden maksimikoko on 40 megatavua, joten harrastajapiireissä liikkuva, kymmeniä pelejä sisältävä noin gigatavun HDS-levykuva ei vielä toimi. Tämä on kuitenkin pieni vaiva, sillä HDF:t on helppo valita valikosta ja ne käynnistyvät automaattisesti haluttuun peliin.

Pääosa kokeilemistani peleistä toimi MiSTerillä oikein mukavasti, vaikka mitenkään sataprosenttisesta yhteensopivuudesta ei voi vielä puhua. Osa ohjelmista jumittui latausruutuun, osassa taas grafiikan tilalla nähtiin pelkkiä häiriökuvioita. Itselleni ennestään tutut pelit tuntuivat hyvin samanlaisilta kuin aidolla X68000 Expertillä – ehkä sillä erolla, että esimerkiksi *Parodius Da!* tuntui hidastelevan hieman vähemmän. Toisaalta käytän oikean koneen kanssa MIDI-ääniä, joita en saanut vielä MiSTeristä ulos, vaikka senkin pitäisi olla mahdollista mt32-pi-projektin avulla. Yhtä kaikki, jos Sharp-peleihin tutustuminen kiinnostaa, tarjoaa tämä ydin siihen aivan kelvollisen ja kynnykseltään alhaisen mahdollisuuden. Toivottavasti kehitystyötä jatketaan, sillä tällaisenaankin ydin on jo varsin käyttökelpoinen.



Aivan kaikki X68000:n pelit eivät vielä suostu yhteistyöhön, mutta mm. Akumajou Dracula, Parodius Da! ja Viewpoint pyörivät oikein mukavasti.

Pasokon-rakkautta

Toinen ilahduttava yllätys oli myös lehdessämme käsitellyn NEC PC-8801:n ilmestyminen MiSTerin Computer-luetteloon. Tämä varhainen Japani-mikro on täkäläisittäin varsin tuntematon, mutta sillekin julkaistiin aikoinaan satoja ellei tuhansia erilaisia pelejä: käännöksiä kolikkopeleistä, porttauksia länsimaisilta mikroilta, Nintendo Famicom -pelien versioita, ihan omia tuotoksia sekä – tietenkin – hakkereiden väsäämiä erilaisia tyttöpelejä. Kaikenlaista hyvin japanilaista siis.

PC-8801 ei ole järin monimutkainen rauta matkittavaksi, joten MiSTer selviää siitä kunnialla jo nykytilassa. Käypää valuuttaa ovat D88-muotoiset levyimaget – laitteessahan on myös mahdollisuus ka-

NEC PC-8801 vaatii tosielämässäkin vähän säätöä, että pelit toimivat. Keskellä Door Door mkII, alhaalla Winter Games.

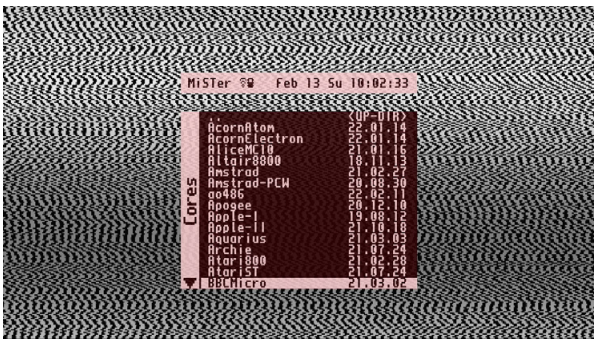
Sega-logo kolmella eri tavalla: ilman jälkikäsittelyä ja PVM- sekä VGA-monitorien shadow maskeilla..

settiaseman liittämiselle, mutta itseltäni puuttuu toistaiseksi tähän liittyvä kalusto ja osaaminen. Tämä ei sikäli haittaa, että materiaalista valtaosa on lerpuilla, vähintään sitten harrastajien itse tekemien kokoelmien osana. Ytimessä on tuki sekä V1- että V2-tilalle ja valinta 4 tai 8 megahertsin kellotaajuudelle. Niitä joutuukin vaihtelemaan, jos haluaa saada mahdollisimman monia pelejä toimimaan. Satunnaisella testailulla kaikki, minkä sain toimimaan oikealla PC-8801mkIISR:llä toimi myös MiSTerillä, ainakin pienen asetusten vaihtelun jälkeen.

Syötä kolikko

Eräs kiinnostava osa-alue, joka on ottanut jättiharppauksia sitten edellisen MiSTer-katsaukseni, liittyy kolikkopeliytimiin. Siinä missä projektin alkuvaiheessa tarjolla oli lähinnä yksinkertaisia 80-luvun alun pelejä, päästään uusimmissa ytimissä jo pitkälle ysärille ja tarjolla on Capcom CPS:n kaltaisia tehokkaita alustoja. Parhaimmillaan pelit toimivat niillä erittäin hienosti, jopa sulavammin kuin esimerkiksi Capcomin omalla Home Arcade -emulaatiokonsolilla.

Ei MiSTeristä kuitenkaan vielä täydeksi MAMEn korvikkeeksi ole, sillä vaikka esimerkiksi Segan System 16 -peleistä osa toimii jo kauniisti, osa (kuten megamakea *Golden Axe*) on tällä hetkellä pelikelvotonta sotkua. Kolikkopeleissä käytettiin monimutkaisia kikkoja kopioinnin estämiseen, ja osaan näistä myös ydinten kehittäjät törmäävät nyt. Tästä huolimatta täysin pelattavia pelejä on jo nyt valtava kasa, ja Jotegon kaltaiset Patreon-rahoitetut kehittäjät puskevat jatkuvasti uutta tavaraa putkeen, joten tulevaisuus näyttää valoisalta.



Tässä näkyy vain osa MiSTerin kattavasta kotimikrovalikoimasta. Valitse mieleisesi ja loikkaa mukaan.

Giga Wing -kolikkopeli pyörii upeasti etenkin huomioiden, että se julkaistiin vasta 90-luvun loppupuolella.

Vielä parempaa on luvassa

Viime aikoina MiSTer-yhteisössä on kuhistu etenkin kahdesta asiasta: CRT-näyttöä matkivista shadow mask -näyttöfiltereistä sekä PlayStation-ytimen ripeästä kehityksestä. Jälkimmäisen kehittäjä **Robert Peip**, joka tunnetaan yhteisössä nimellä **FPGAzumSpass**, aloitti omien sanojensa mukaan FPGA-kehitystyön harrastuksena, teki sitä sitten työkseen vuosikaudet ja on nyt palannut harrastuksen pariin – koska se on hauskaa. Lähes 900 Patreon-tukijaa ainakin auttaa myös taloudellisesti.

Jätän PlayStation-tutkimukset tuonnemmaksi, sillä yhteensopivuus on tällä hetkellä vain osittaista ja uusimman version hankkiminen taas hieman hankalaa, koska keskeneräinen ydin ei ole vielä update_allin piirissä. Se on kuitenkin jo nyt ylipäätään käytettävissä yhdellä SDRAM-moduulilla, kun aiemmin vaadittiin kaksi, ja toimivia pelejä on useita. Palataan tähän siis joskus myöhemmin.

Eikä tässä ole tietenkään kaikki, sillä MiSTer taipuu vaikka mihin: se voi olla 486-pohjainen MS-DOS-kone, Amiga, Atari ST tai Acorn Archimedes, Colecovision, Super Nintendo tai Sega Mega-CD, melkein mikä tahansa laite noin vuoteen 1995 mennessä. Se tekee erilaisiin järjestelmiin tutustumisesta todella vaivatonta – jopa verrattuna tietokoneella toimiviin emulaattoreihin. Laitteen yhteisö on erittäin aktiivinen ja uusia asioita tapahtuu aivan jatkuvasti, kun eri projekteissa tehtävät läpimurrot tukevat toisiaan. Tarjolla on jopa monia täysin yhdysrakenteisia toteutuksia, joihin pitää vain lisätä DE10-Nano keskusyksiköksi, sekä sovittimia klassikkokonsoleiden ohjaimille.

Ainoa merkittävä ongelma tällä hetkellä onkin, että valmistuskapasiteetti- ja logistiikkaongelmien vuoksi itse DE10-Nanon saatavuus on surkea ja kortteja kaupitellaan jopa moninkertaisilla hinnoilla alkuperäiseen nähden. Tämä vie valitettavasti pontta siltä ajatukselta, että järjestelmän on tarkoitus olla edullinen johdatus FPGA:n maailmaan. Epäselvää on myös, kuinka kauan Intel jatkaa DE10-Nanon myynnin tukemista oman FPGA-kehitysohjelmansa osana, sillä korttia on myyty jo pitkään huomattavan edullisesti sen kykyihin nähden. Hyvä puoli on se, että tehty työ ei menisi täysin hukkaan vaikka laitealustaa vaihdettaisiinkin, mutta DE10-Nanon ympärille syntynyt lisävarustemarkkina saa toivomaan, että sillä on vielä vuosia edessäänkin.

EVERCADE CODEMASTERS COLLECTION 1

Amigaa taskuun? Ehkä vähän sinne päin

Vanhoja kunnon pelimoduuleja käyttävä Evercade-käsikonsoli julkaistiin vajaa pari vuotta sitten vetelemään nostalgianaruista (Retro Rewind 3/20). Samaa vetelyä jatketaan viime vuoden lopulla julkaistulla Codemasters-kokoelmalla. EA:n äskettäin hankkima Codemasters on nykyään tunnettu lähes yksinomaan autopeleistään, mutta codiet olivat monipuolisen pelivalikoimansa myötä kotitietokoneharrastajille ja osittain myös konsolipelaajille hyvin tuttu nimi jo 80- ja 90-luvuilla.

Tämä kokoelma ei kuitenkaan keskity pelkästään Codemastersin aikanaan tekemiin peleihin, vaan myös pelisarjoihin, jotka ovat tavalla tai toisella päätyneet saman sateenvarjon alle. Isoimmat retrovibat niistä herättää Sensible Software, jonka kanssa Codemastersit pistivät hynttyyt yhteen vuonna 1999. Tästä yhdistelmästä koostuukin sitten 17 pelin paketti, joka saa ainakin kasuaalien Amiga-fanien sydämet hyristelemään.

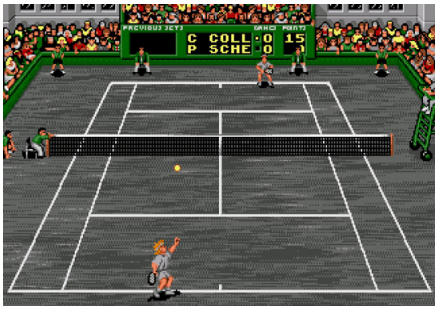
Pelimoduulin kansikuvan tähtinä ovat ne varmasti jokaisen Amiga-harrastajan kokoelmasta (vähintään piraattiversioina) löytyneet *Sensible Soccer*, *Cannon Fodder* ja *Super Skidmarks*. Mukana on toki myös vähän harvinaisempaa tavaraa, kuten vain Englannissa Mega Drivelle (muualla PC:lle) julkaistu *Psycho Pinball*, kaksi kappaletta luolamies Big Nosen seikkailuja sekä useita pelejä NESille ilman Nintendon siunausta julkaistuilta Quattro-kokoelmilta. Kyytiin on saatu myös julkaisematon *Tennis All Stars*. On hieman erikoista, ettei mukana ole yhtään Codemastersin maskotiksikin nousseen Dizzy-hahmon peliä, mutta ehkä niitä säästellään sitten seuraaviin kokoelmiin.

Tämän pelikokoelman arvostelu piti tulla jo viime Retro Rewindiin, mutta ainakin osa ensimmäisen erän pelimoduuleista oli viallisia, joten niiltä käynnistyivät vain 8-bittiset pelit. Tilalle lähetettiin uusi, ja vaikka tulliviranomaisten suuntaan lähti nyt sitten tuplamaksut, onpa minulla myös se "harvinainen""toimimaton" versio hyllyssäni.

Aloitetaan siitä tunnetuimmasta Amiga-kolmikosta – tai no, Mega Drive -versioitahan nämä ovat. Alkuperäisalustan puutteesta johtuen missataan muun muassa *Cannon Fodderin* mainio alkulaulu, mutta muuten pelikolmikko on kolmikerroshampurilainen nostalgian nälkäiselle. Varsinkin tykinruokailu on edelleen toimivaa viihdettä, johon syntyy "vielä kerran" -koukku. *Super Skidmarksin* ohjastamiseen ottaisin kyllä mieluummin vanhan kunnon joystickin ja *Sensible Soccer* on parasta kaveria vastaan.

Amigaviboja voi saada myös reaaliaikastrategia *Mega-Lo-Maniasta*, joka kyllä julkaistiin monille alustoille, sekä hieman yllättäen *Psycho Pinballista*, joka ei Amigalle saapunut aikoinaan lainkaan. Se vain on erinomaisen hyvin tehty klooni *Pinball Dreamsista*. Niin hyvä, että se voisi käydä sellaisenaan lisäkentistä enkä epäilisi autenttisuutta. Ehdin tekemään hatunsyöntilupauksen tekijäporukoiden yhtäläisyyksistä, mutta niitä ei löytynytkään, joten pieleen meni. Väittäisin, että *Psycho Pinball* pelikenttineen ja bonuspeleineen on pelattavuudeltaan jopa niin hyvä, että se yksinään perustelee koko kokoelman ostamisen. Useamman kerran teki mieli tarttua uudelleen Evercadeen pistämään taas kerran high-scorea uusiksi.

Kasibittisten pelien valikoimasta ei sitten suurta hurrattavaa löydykään: enimmäkseen vain yksinkertaisia perustasoloikkia ja räiskintöjä. Massasta nousi ainoastaan *The Ultimate Stuntman*, joka on aika hämmästyttävän monipuolinen teos. Pelistä löytyy niin ylävinkkelistä ajelua, sivuvinkkeli-run'n'gunia, shoot'em'uppia, kiipeilyä kuin pomminpurku-puzzleakin. Alkufiilikseni oli hämmästyneen vaikuttunut, mutta monipuolisuudestaan huolimatta *TUS*:iin turhautuu nopeasti – jos vaikka peli osaakin vähän kaikkea, se ei ole mestari missään.



Evercadella on ollut tapana pistää kokoelmille aina myös joku härömpi tapaus ja tällä kertaa se on *Tennis All Stars,* joka jäi ysärillä julkaisematta. Teknisesti peli vaikuttaa oikein valmiilta, ja valmis se olikin. Peliä vain lähdettiin kehittämään hieman eri suuntaan, kun mukaan saatiin tennistähti **Pete Sampras**. Helppoa sen pelaaminen pikkuruudulta ei ole, vaikka pidänkin itseäni tennispelien veteraanina. Tässä pelissä vilahtaa myös se kokoelmalta kateissa ollut Dizzy. Pelin mielenkiintoisinta antia on Crazy Tennis -pelitila, jossa kentällä tapahtuu monenlaista ylimääräistä.

Kokoelmana *CC#1* on vähän epätasainen, mutta ymmärrän ettei kaikkia hittejä voi pistää yhteen kasaan, jotta myöhemminkin olisi julkaistavaa. Hinta (21,50 eur) ei kuitenkaan ole paha. Tasku-Amigana se ei oikein palvele, varsinkaan jos on puristi peliversioiden suhteen, mutta *Psycho Pinball*, voi pojat, *Psycho Pinball*.

RUNESCAPE: THE FIRST 20 YEARS – AN ILLUSTRATED HISTORY (DELUXE EDITION)

Jukka kirjoitti kaksi vuotta sitten Retro Rewindissa otsikolla Sitkeät sissimorpit MMORPG-peleistä, jotka eivät suostu kuolemaan, vaikka kilpailijoita tulee ja menee. Yksi tervaskanto on RuneScape, jonka "Old School" -versio täytti viime vuonna jo 20 vuotta. Kaikkine versioineen sillä on yhä miljoonia pelaajia ja erityisesti tälle uskolliselle fanikunnalle – sekä niille, jotka ovat näiden vuosien aikana Runessa aikaansa viettäneet – julkaistiin viime vuonna komea juhlateos.

Teksti: Manu Pärssinen

Hieman uskaliaasti nimetty *RuneScape: The First 20 Years* kertoo pelin tarinan vuodesta 2001 nykypäivään runsaalla kuvituksella varustettuna. Arvosteluun saapui perusversiota yllättävänkin paljon (50 euroa) kallimpi *Deluxe Edition,* joka onkin tuhti paketti. Vahvan erikoiskansion sisuksissa on paksuilla kansilla varustettu 224-sivuinen kirja sekä kaksi taidetulostetta sisältävä pienempi kansio. Deluxe-versiossa kirjan sivujen reunukset on kullattu ja muutenkin kullalla viimeisteltyjä yksityiskohtia löytyy paketista sieltä täältä. Kokonaisuus huokuu luksusta ja viimeisen päälle viilattua painotyötä. Se onkin rajoitettu painos, joka on tarkoitettu keräilyesineeksi.

Myönnän, että itse en ole *RuneScapen* syvään suohon uponnut: vietin siellä 2010-luvun alkupuolella muutamia viikkoja tutustuen sekä vanhaan että uudempaan versioon. En ole kuitenkaan sydämeltäni MMO-pelaaja, joten kokeiluksi nuo jäivät. Tämän kirjan lukemisen kannalta sillä ei kuitenkaan ole kovin isoa merkitystä. Pelin veteraanit saavat varmasti enemmän irti monenmoisista viittauksista ja muistelevat haikeina kirjassa mainittuja tapahtumia, *RuneScape*-nuubit taas pääsevät tutkimusmatkalle tuntemattomaan. Kuinka tämä peli on oikein selviytynyt menestyksekkäästi hengissä näistä 20 vuodesta, joissa on kuitenkin ollut monenmoista vuoristorataakin?

Kirjan on hartaudella koostanut brittiläinen pelijournalisti **Alex Calvin**, joka on uransa varrella kirjoittanut useimmille isoista pelisivustoista. Teos ei ole niin "sohvapöytäkirjamainen" kuin viime lehdessä arvosteltu *Pac-Man*-historiikki, vaan tuntuu enemmän tietokirjalta – luettavaa on paljon, vaikka komeita kuviakin on toki mukaan laitettu roppakaupalla.

Tietokirjamaisuus ei kuitenkaan tarkoita, etteikö *RuneScape*-juhlateosta olisi viihdyttävä lukea. Pienten alkupuheiden jälkeen lähdetään liikkeelle koko pelin syntytarinasta - ja kolmesta **Gowerin** veljeksestä. Historian kerrontaa katkotaan retkillä pelaajille tuttuihin maisemiin, hahmoihin sekä vihollisiin *RuneScapen* maailmassa, ja juuri näissä kohdin vanhan koulun pelaajien silmät varmasti kostuvat. Luettavaa tarjotaan siis tasapuolisesti niin muistelijoille kuin uusille tutustujillekin.

Tarina jatkuu pelin julkaisuhetkistä kertoen ja hauska anekdootti on muun muassa syy siihen, miksi se ensimmäinen pelipalvelin ei pärjännytkään niin hyvin kuin oli suunniteltu. En kuitenkaan spoilaa sitä tässä. Vähitellen Jagex yrityksenä kasvoi pelin suosion myötä ja vaikka kaoottisia vaiheita oli useampiakin, *Runen* menestys jatkui vuodesta toiseen. *RuneScape 2* toi peliin useita uudistuksia ja uusia pelaajia, mutta vanhakin versio pidettiin hengissä. Ehkä tällainen OG-pelaajakunnan kunnioittaminen on yksi syistä siihen, että MMO-tervaskanto ei peruskalliostaan hevillä irtoa.

Toinen syy saattaa olla se, minkä **Neil "Mod Neil" Richards** kirjassa toteaa: Jagex ei koskaan edes halunnut kilpailla isojen pelitalojen kanssa, koska talossa tiedettiin, ettei niitä voitaisi päihittää. Sen sijaan pelissä keskityttiin hauskoihin ja hupsuihin juonenkäänteisiin sekä yksityiskohtiin; parodiointiin, joka ei monessa vakavamielisessä, isossa MMO-pelissä olisi mahdollista.

Kirjan neljäs luku käsittelee niitä ongelmia, joita syntyi häirikköpelaajista, boteista, kultafarmauksesta sekä oikealla rahalla käydystä pelitavaroiden kaupasta. Näitä jokaisen verkkopelin vitsauksia vastaan taisteltiin Jagexillakin monin erilaisin tavoin kissa-hiiri-leikissä, eivätkä ideat vastatoimenpiteiksi olleet aina järkevimmästä päästä, mikä nopeasti opittiin. Myöhemmissä luvuissa kerrotaan sitten vielä pelin sekä emofirma Jagexin kehitysvaiheista nykyhetkiin asti.

Itselleni juhlateoksen alkupuoli oli mielenkiintoisempaa luettavaa, vaikka eri kehitysvaiheista ja -versioista kertoviin lukuihin oli aina ripoteltu pieniä makupaloja tapahtumista kulisseissa. Kuten kuuluisikin, osa kirjasta on omistettu myös pelaajille: heille, jotka käyvät Runefestissä tapaamassa toisiaan, cosplayaavat ja toivottavat tervetulleiksi uudet pelaajat. "There is no doubt, that *RuneScape* has succeeded for 20 years because of its players", kuten **Dave "Mod Osborne" Osborne** aivan viimeisillä sivuilla toteaa.

Toki kirja on kirjoitettu Jagexin näkökulmasta, joten kovin karskia omaan nilkkaan osuvaa kritiikkiä ei sivuilla esiinny, vaikka tehtyjä virheitä toki myönnetään. Itseni yllätti se, kuinka paljon rautalankavirityksiä, työtä ja sisältöä *RuneScapen* eri versioissa on vuosien varrella oikeasti ollut. Vaikeahan tulevaisuutta on ennustaa, mutta ehkä 20 vuoden päästä nähdään tuplasti paksumpi kirja, jonka nimenä on "RuneScape: The First 40 years".

RuneScape:
The First 20 Years
Alex Calvin, Jagex, 2021
runescape.backstreetmerch.com
85 eur (Deluxe, perusversio 35 eur)

TORVISOVITUKSIA JA DOOM-TYYLISTÄ JYTKETTÄ

RETROPELIMUSIIKKI RAIKASI MYÖS VUONNA 2021

Retro Rewindin musatoimittaja valitsi neljä vuoden 2021 kohokohtaansa retropelimusiikin rintamalta. Esittelyssä ovat vuoden eeppisimmät uutuudet, erikoisin löytö ja kuunnelluin retropelilevy.

Teksti: Miika Auvinen

Myös vuonna 2021 elämääni riemastuttivat vanhempien pelien musiikeista ammentavat coveralbumit. Nyt onkin aika puida vuoden kohokohtia retropelimusiikin tiimoilta. Esittelen kaksi parhaiten mieleen jäännyttä vuonna 2021 retropelimusiikista uudelleenversiointeja julkaissutta kokoonpanoa. Kohokohtien ”erikoisin löytö” on vuodelta 1999 ja kuunnelluin albumini taas vuodelta 2014, eli kaikki artikkelin albumit eivät ole uunituoreita.

Vuoden 2021 tragedia koskee Materia Collective -pelimusiikkikollektiivin kohtaloa. Kollektiivi oli julkaissut varsin monipuolisen kattauksen eri tyyleillä tehtyjä uudelleenversiointeja retropelien soundtrackeista, ja parhaimmillaan kollektiivi oli *Final Fantasy* -sarjan eri osien ja *Xenogearsin* soundtrackien uudelleenversioinneissa. Kysyin kollektiivin mediavastaavalta, miksi albumit ovat kadonneet – syynä ovat laajan tekijäjoukon tekijänoikeudet. Toivottavasti ryhmän levyt tulevat joskus vielä jonnekin laillisesti kuunneltavaksi!

Vuoden eeppisimmät biisit

Pelimuusikko **Mick Gordon** on taikonut *Doom Eternal* -soundtrackiin uniikin äänimaailman. Geoffplaysguitar -projektin parhaissa biiseissä tämä modernien *Doomien* äänimaailma siirretään Final Fantasy -klassikoihin. *Final Fantasy VI:n Dancing Mad* ja *Final Fantasy VII One Winged Angel* -lopputaistelubiisit saavat osakseen uutta jylhää sähäkkyyttä, ja elektroninen demonintappovyörytys sopii hyvin myös *Final Fantasyjen* enkelintappobiiseihin.

Geoffplaysguitar on kitaristi ja säveltäjä **Geoffrey Dayn** sivuprojekti, ja vuonna 2021 ilmestyi coverbiisejä myös *Bloodbornesta, Halosta, Nier*-sarjasta ja monista muista. Hieman samantyylistä eeppistä pelimusatunnelmointia esittää myös FalKKonE nimellä julkaiseva **Michał Sokołowski** .

Miten en ollut kuullut tästä aiemmin?

The Game Brass -puhallinsoitinorkesteri tekee torvicovereita klassisista videopelimusiikeista. Monella levyistä on oma teemansa, kuten kauhupelit tai lumi- ja linnakentät. Osa albumeista keskittyy vain yhteen peliin tai pelisarjaan. Orkesteri julkaisi vuonna 2021 kaksi *Zelda*-teemaista levyä, ja torvisoitto sopii odottamattoman hyvin pelisarjan melodioihin.

Parhaimmillaan yhteen musiikki on riemukasta torvitunnelmointia, mutta pettymyksiäkin sovitusten joukosta löytyy, kuten mainitun kauhupelilevyn *Silent Hill 2:n Laura's Theme* ja *Final Fantasy VI:*n kummitusjunan taustamusiikin hieman lässähtäneet versioinnit. Yhtyeen torvensoittajista moni on opiskellut soittimensa tiimoilta yliopistossa tai muussa korkeakoulussa, joten amatöörejä jäsenet eivät ole.

Erikoisin suoratoistolöytö

Vuonna 1999 ilmestyi *Um Jammer Lammy,* rytmipelien kuningatar. Pelissä kitarasankari Lammyllä on kiire soittamaan kitaraa MilkCan-yhtyeen riveissä. Lammy selviää matkan varrella kinkkisistä tilanteista visualisoimalla kitaransoittoa, kun oikeasti tarkoituksena olisi nukuttaa sylissä oleva vauva tai lentää lentokonetta.

Vuoden 2021 yllättävin löytö olikin suoratoistopalvelujen uumenista löytyvä oheistuote, MilkCan-yhtyeen muka soittama levy *Make It Sweet!*. Levy on yhtä sekava ja energinen kuin Lammyn peli. Kun pelissä vokaalit ja kitaransoitto vuorottelevat tönkösti, levyn sovituksissa mukana on ihan mukiinmenevääkin poppia. Vuonna 1999 julkaistu levy on pelimusamaailman anomalia, outo erikoisuus, ja sen löytyminen esimerkiksi Spotifystä on pieni ihme.

Kuunnelluin retropelicover-levy

Spotify listaa vuoden lopussa käyttäjilleen kyseisen ajanjakson kuunnelluimmat biisit, levyt ja artistit. Kuunnelluin levy kohdallani oli vuoden 2014 Megadriver-yhtyeen *Double K.O.* Levy keskittyy klassisiin tappelupeleihinm ja sitä kuunnellessa utuiset aamuherätykset ja ankeat loskakelit täyttyvätkin energiasta.

Mukana on metallisovituksia pelisarjoista kuten *Street Fighter, Mortal Kombat* ja *Soul Edge*. Omat suosikkini levyltä ovat *Fatal Fury* -sovituksia, ja varsinkin Big Bear -hahmon teemabiisi *Choke Hold* laittaa adrenaliinit liikkeelle. Osa biiseistä on laulettuja: vokaalit ovat epävireisiä ja räkäisiä, mutta niissä on hyvää tunnelmaa. Uudemmilla Megadriverin levyillä on ollut aiheina muun muassa shoot ‘em up -pelit ja *Megaman X* -sarja.

MANUN KUMMAT VEHKEET

Teksti: Manu Pärssinen

Kuvat: Blake Patterson / Wikimedia Commons, Manu Pärssinen

Lähes kaikilta pelikeräilijöiltä löytyy komeroistaan Commodorea tai Nintendoa, mutta paljon on niitäkin, jotka haalivat monille kummallisempaa kamaa. Keräilijäryhmä Pelikonepeijoonien Manu otti jälleen kokoelmastaan tutkailtavaksi vähän oudomman laitteen.

SAMSUNG NUON DVD-N501 (JA KUMPPANIT)

Aikakausi 2000–2001 oli ehkä huonoimpia hetkiä hyökätä mukaan kilpailemaan konsolimarkkinoista. PlayStation 2 jylläsi, Microsoftin Xboxia sekä Nintendon Gamecubea odoteltiin innolla ja Segan kohtalostakin olisi voinut ottaa oppia. Se ei kuitenkaan pelottanut kalifornialaista VM Labsia, olihan sen perustajalla kokemusta menestyskonsolista: Atari Jaguarista.

No kyllähän se tiedetään, että Atarin Jaguar floppasi. VM Labsin **Richard Millerillä** oli uuden firmansa kanssa kuitenkin hyvä pohja-ajatus: vakaa usko siihen, että kohta joka kodissa olisi DVD-soittimia. Jos siihen markkinarakoon ymppäisi pelikonsolin, voisi alkaa rakentaa rahasäiliötä! Tuloksena oli lopulta Nuon, DVD-soittimiin sisällytetty peliteknologia. Jaa että ette ole kuulleet siitä? Ei mikään ihme.

Miller osasi hankkia yhteistyökumppaneita, ja Atarin entisten työntekijöiden lisäksi mukaan houkuteltiin muun muassa Motorola ja Toshiba. "Project X:ksi" nimettyä teknologiaa esiteltiin sekä elokuvastudioille että pelistudioille vaisuin tuloksin, mutta maaliin asti lopulta kuitenkin päästiin. Pelipuolella lippulaivana oli **Jeff Minterin** *Tempest 3000*, eli jatko-osa Jaguarille julkaistulle *Tempest 2000*:lle.

Nuonin tekijöiden haaveena oli todellakin päästä osaksi jokaista DVD-soitinta, eli periaatteessa siinä oli sama perusajatus kuin MSX:llä, Philipsin CD-i:llä ja 3DO:lla – yhteinen alustateknologia, jota hyödyntäisivät monet eri laitevalmistajat ja VM Labs saisi aina oman siivunsa. Nuon-yhteensopivia soittimia saivat kauppoihin asti Samsung, RCA ja Toshiba. Motorola käytti samaa teknologiaa Streamaster-digiboksissaan.

Teknologia ei kuitenkaan ottanut tuulta alleen eikä saanut lisätukea muilta DVD-soitinten valmistajilta tai pelinkehittäjiltä. VM Labs haettiin konkurssiin vuonna 2001, paketoitiin lopullisesti vuonna 2002 ja pahnan pohjimmainen Nuon-yhteensopiva DVD-soitin rullasi tehtaasta vuoden 2004 syksyllä. Ainakin joidenkin lähteiden mukaan viimeinen niitti oli Disneyn elokuvapuolen kanssa käytyjen neuvottelujen kariutuminen.

Nuonista kerrottiin myös Skrolli-lehden kaikkien aikojen ensimmäisessä numerossa otsikolla *"EI NÄIN: CD-i, 3DO ja Nuon"*. Siinä **Mikko** toteaa, että yksi epäonnistumisen syy saattoi olla myös epäselvä brändäys. Pelkkä pieni Nuon-logo DVD-soittimelta näyttävässä DVD-soittimessa ei kertonut ostajalle, että tässä ollaan nyt pelikonsoliakin hankkimassa. Se isoin syy taisi kuitenkin olla aikaisemmin julkistettu PlayStation 2 DVD-asemineen.

Mitä Nuonin lisärauta sitten tarjosi?

Standardin epämääräisyyden vuoksi tarjonta vaihteli. Joissain malleissa laitteen ytimessä sykki aikaisekseen kohtuullisen kyvykäs Aries 3 -siru, joissain taas sen heikompi edeltäjä. Piiri perustui 128-bittiseen nelikanavaiseen, rinnakkaisuutta hyödyntävään VLIW-prosessoriarkkitehtuuriin. Se hallinnoi keskusyksikköä (Nuon MPE 54 Mhz tai 108 Mhz), prosessoi kuvaa ja ääntä sekä osasi erilaisia videoskaalauskikkoja, äänisynteesiä ja kehnoa 3D-peligrafiikkaa. Laitteessa oli 32 megatavua keskusmuistia sekä puoli megatavua äänimuistia, ja videosuorittimessa lisäksi kahdeksan megatavua kuvamuistia.

Rinnakkaistodellisuudessa kaikki olisi voinut loksahtaa kohdalleen. Elokuvateollisuus olisi voinut ihastua lisäominaisuuksia DVD-elokuville tarjoavaan vimpaimeen, mutta todellisuudessa sellaisia elokuvia julkaistiin vain neljä. Yksi niistä oli *Planet of the Apes,* jossa Nuon-raudalla pystyi mm. zoomailemaan tiettyihin asioihin, katselemaan diaesitystä elokuvan tekemisen vaiheista sekä valitsemaan helpommin, mihin kohtaan elokuvaa haluaa loikata.

Rinnakkaistodellisuudessa peliteollisuus olisi voinut ihastua joka kodin DVD-soittimessa lymyilevään pelikonsoliin, mutta todellisuudessa pelejä julkaistiin vain kahdeksan. *Tempest 3000*:n lisäksi laitteelle ilmestyi *Ballistic*- ja *The Next Tetris* -puzzlepelit, FMV:ltä tuoksuva *FreeFall 3050 A.D.* -toimintapeli (joka löytyy myös Steamista), *Iron Soldier 3* -mechapeli, *Merlin Racing* -kart-ajelu sekä *Space Invaders XL*. *Tempest* niistä rankataan yleensä parhaimmaksi, enkä väitä vastaan. *Iron Soldier* vedettiin pois myynnistä, koska se ei toiminut joidenkin soitinten kanssa.

Se kahdeksas peli oli ainoastaan Etelä-Koreassa julkaistu tasoloikka *Crayon Shin-Chan*, joka oli lukittu aluekoodilla monien DVD:iden tapaan. Sitä pystyi pelaamaan ainoastaan Etelä-Koreassa julkaistulla Samsung DVD-N591 -laitteella, jonka mukana peli tulikin. Täyttä varmuutta ei ilmeisesti ole siitä, julkaistiinko Nuon maassa virallisesti vai oliko kyseessä vain koe-erä. *Crayon Shin-Chan* saattaakin olla siis yksi maailman harvinaisimmista pelikokemuksista.

Eri Nuon-laitteita julkaistiin yhteensä kahdeksan mallia: Pohjois-Amerikassa kaksi Samsungia ja kaksi RCA:ta (jotka eivät tukeneet julkaistuja pelejä) sekä yksi Toshiba. Lisäksi kauppoihin asti ehti kaksi Samsungin laitetta Euroopassa, sekä se yksi edellä mainittu Koreassa. Peliohjaimia oli neljää mallia: Samsungin oma, pari erilaista HPI-nimiseltä valmistajalta (yläkuvassa oleva Nintendo 64 -ohjaimen näköinen vekotin) sekä yksi vähän yllättäen Logitechilta. Kun kaikkien usko projektiin alkoi hiipua, Samsung poisti uudemmista malleistaan (Eurooppa, Korea) peliohjainliitännän. Osaa peleistä pystyi pelaamaan huonosti myös laitteen mukana tulleella kaukosäätimellä.

Aina löytyy harrastajia

Kuten monien altavastaaja-alustojen kanssa, Nuonillekin syntyi innokas harrastajajoukko, jonka kädenjälkeen voi käydä tutustumassa mm. Nuon-Dome-sivuston kautta (nuon-dome.com). Laitteelle on julkaistu yli 30 kotikutoista pelituotosta ja harrastajat ovat kaivaneet esiin myös julkaisematta jääneitä prototyyppipelejä. Yksi niistä on *Bust-A-Move 4*.

Harrastamista rajoittaa muun muassa peliohjainten harvinaisuus. Tätä kirjoittaessa eBaylla ohjaimesta voi joutua pulittamaan jopa enemmän kuin itse laitteesta, jos sellaisia sattuu edes myynnissä olemaan. Kotitekoisesta ohjaimesta on ollut puhetta harrastajapiireissä vuosien ajan, mutta ongelmallista on, että ohjain vaatii erityisen piirin toimiakseen. Toinen mutka on se, että Nuon-laitteista vain tietyillä malleilla (RCA:t, Samsungin N50x:t) voi ajaa homebrew-ohjelmia CD-R-levyltä.

Jeff Minter teki muuten laitteelle muutakin kuin taas yhden *Tempestin*. "*VLM*" on kirjainyhdistelmä, jonka jotkut saattavat tunnistaa Jaguar CD:ltä. *Virtual Light Machine* oli "valosyntetisaattori", jollaisten kanssa Minter oli leikitellyt jo 80-luvulla. Ohjelma siis loi ruudulle erilaisia psykedeelisiä kuvioita musiikin kuvitukseksi. Sellaisen versio *VLM-2* oli sisään rakennettuna Nuonissa, mutta Nuonille perinteisesti tässäkään ei voitu mennä yksinkertaisesti: jotkut laitteet tukivat yli 150:tä efektiä, toiset taas vaikkapa vain kahdeksaa. Jotkut osasivat reagoida musiikin rytmiin paremmin, toiset huonommin.

Ehkä Nuonin epäonnistumisen syy olikin yksinkertaisesti se, että se oli kamala sotku.

Nuonin 3D-kyvyt olivat jostain ensimmäisen ja toisen PlayStationin välimaastosta. Kuvassa Merlin Racing.

PEIJOONIMUISTO

Olin tietoinen Nuonista ja sen epämääräisyydestä jo laitteen julkaisun aikoihin, mutta eipä niitä (ilmeisistä syistä johtuen) koskaan tänne asti rantautunut. Se kyllä vähän harmittaa, ettei tullut haalittua Nuon-ohjaimia silloin kun ne olivat verkkokaupoissa 20–30 euroa.

sendanor.fi

Luotettavia hosting-palveluita jo 15 vuotta

Maksa kerran, käytä 10 vuotta!

...tai tilaa lyhyemmillä kausilla jälleenmyyjiltämme

10v

Virtuaalipalvelimet

Levytila	Muisti	CPU	Verkko	Hinta
100 GB	1 GB	1x 1/10	100 Mbps	299 €
1 TB	1 GB	1x 1/10	1000 Mbps	499 €
4 TB	4 GB	2x 1/5	1000 Mbps	1699 €
12 TB	16 GB	4 x 1/5	1000 Mbps	9999 €

Sisältää pääkäyttäjän oikeudet, IPv6-osoitteen, RAID1-levytilan, varmuuskopioinnin, tuen ajaa kontteja ja yhteensopivan käyttöjärjestelmä (esim. Ubuntu, Debian, CentOS, Arch, NixOS tai FreeBSD)

IPv4-osoite 100 €, IPv6-verkko 20 €, 10 Gbps 20 €/tunti tai 1000 €/kk, ylläpitotyöt 124 €/tunti, varmuuskopiopalautukset 100 €

Tutustu ja tilaa kotisivuiltamme!

Koodilla **SKROLLI** **10% alennus!**

10v

Webhotellit

20 GB levytilaa
10 tietokantaa
10 tunnusta

249 €/10 vuotta

10v

Sähköposti

10 GB levytilaa
10 laatikkoa
100 aliasta

99 €/10 vuotta

10v

Shell-tunnus

1 GB levytilaa
1 SSH-tunnus
1 IPv6-osoite

25 €/10 vuotta

Hinnat sisältävät arvonlisäveron (24%) ja palvelut vähintään kymmeneksi vuodeksi Tarjoukset voimassa 31.12.2022

sendanor.fi